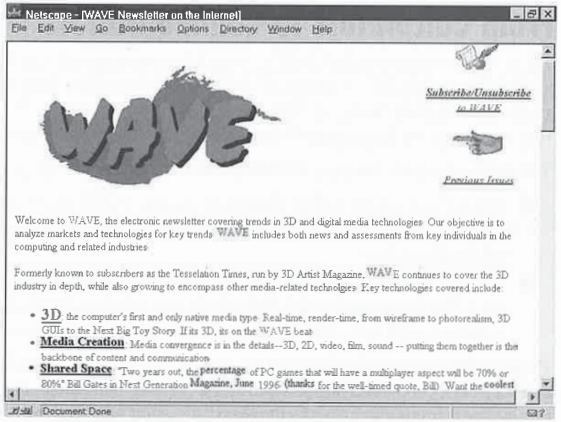

FIGURE 41.1. *HTML's raison d'etre is to provide nicely formatted information that is easy to navigate.*

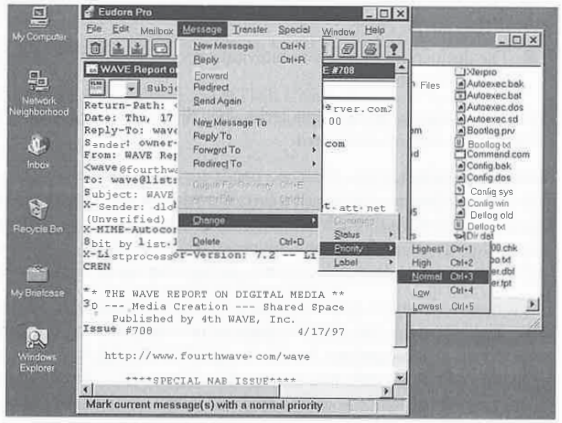

FIGURE 41.2. *Today's computer interfaces are simply too confusing and hard to learn for many of tomorrow's applications. (Note that the actual information content shown here is the same newsletter as in Figure 41.1.)*

From the Desktop to the Conference Table

When personal computers first started appearing on desks, people bought them to help with the kind of tasks that take place on a desktop: reading, writing, calculating, and shuffling pages of information around. This reflected the business model of the mid-20th century: Isolated

workers and companies spent most of their time processing information in their own offices and a much smaller portion of their time directly communicating that information between individuals and companies.

As the century draws to a close, that business model is showing severe signs of strain. More and more time is spent in communication, and less and less in isolation. Workers who were trained in the older way of doing business complain that they now spend so much time attending meetings and conferences, or answering voice mail and e-mail, that no time is left to "get any work done." Those from the newer generation of employees and entrepreneurs, on the other hand, complain when coworkers or business associates create communication bottlenecks by being "out of touch" and "hard to reach."

The old desktop metaphor is increasingly useless for doing the most crucial tasks in a communications-oriented business world. As the personal computer becomes more of an interpersonal communicator, the interface must be more of a conference table than a desktop. Like a conference room, your computer is now primarily a place to exchange textual and graphical documents, audiovisual presentations, and verbal interaction. Unlike most conference rooms, your computer allows you to directly exchange information with hundreds of millions of people all over the world.

HTML as the New User Interface

As the role of the computer evolves, HTML is becoming more and more central to nearly everything we do with computers. HTML is *the* global standard for connecting all types of information together in a predictable and presentable way.

HTML provides a painless and reliable way to combine and arrange text, graphics, sound, video, and interactive programs. And unlike older proprietary page layout standards, it was designed from the start for efficient communication between all kinds of computers worldwide. At this point, the likelihood of any other standard unseating it as king of the communications hill seems remote.

The prominence of HTML, however, does *not* mean that Web browsers will be a major category of software application in the coming years. In fact, the Web browser as a distinct program has already nearly disappeared. Microsoft Internet Explorer 4.0, for instance, does much more than retrieve pages from the World Wide Web. It lets you use HTML pages as the interface for organizing and navigating through the information on your own computer, including directory folders and the Windows desktop itself. In conjunction with HTML-enabled software such as Office 97, HTML becomes the common standard interface for word processing, spreadsheets and databases as well. The new Netscape Communicator 4.0 is also much more than a Web browser. It uses HTML to integrate all types of media into e-mail, discussion groups, schedule management, business documents, and collaborative project management. (See Figure 41.3.)

IPR2024-00143
Apple EX1011 Page 435

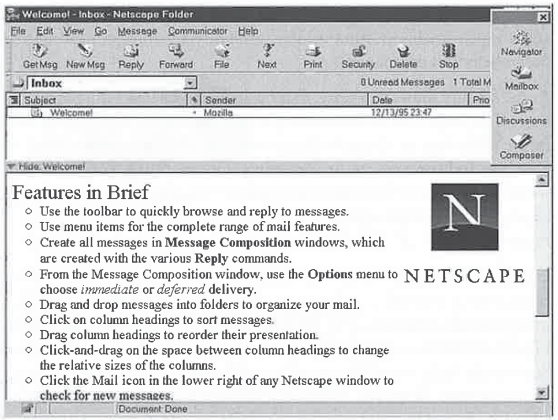

FIGURE 41.3. *Netscape Communicator 4.0 offers a vision of the future by integrating HTML into e-mail, discussion groups, and other business communication.*

Meanwhile, HTML support is being included in every major software release so that every program on your computer will soon be able to import and export information in the form of HTML pages. In a nutshell, HTML is the glue that holds together all the diverse types of information on our computers and ensures that it can be presented in a standard way that will look the same to anyone in the world.

In a business world that now sees fast, effective communication as the most common and most important task of its workers, the information glue of HTML has the power to connect more than different types of media. It is the hidden adhesive that connects a business to its customers, and connects individual employees to form an efficient team. Knowing how to apply that glue—the skills you gained from this book—puts you in one of the most valuable roles in any modern organization.

The Digital Media Revolution

The most important changes in the next few years may not be in HTML itself, but in the audience you can reach with your HTML pages. Many Web site developers hope that Internet-based content will have enough appeal to become the mass-market successor to television and radio. Less optimistic observers note that the global communications network has a long way to go before it can even deliver television-quality video to most users.

I won't pretend to have a magic mirror that lets me see how and when HTML becomes a mass-market phenomenon. But one thing is certain: All communication industries, from television to telephony, are moving rapidly toward exclusively digital technology. As they do so, the lines between communication networks are blurring. New Internet protocols promise to optimize

multimedia transmissions at the same time that new protocols allow wireless "broadcasters" to support two-way interactive transmissions. The same small satellite dish can give you both Internet access and high-definition TV.

Add to this trend the fact that HTML is the only widely supported worldwide standard for combining text content with virtually any other form of digital media. Whatever surprising turns the future of digital communication takes, it's difficult to imagine that HTML won't be sitting in the driver's seat.

More than a million people can already access the Internet without a "real computer" through TV set-top boxes and from WebTV Inc., cable TV companies, and digital satellite services. These devices are only the first wave of much more ubiquitous appliances that provide HTML content to people who wouldn't otherwise use computers.

NOTE

The prospect of mass-market HTML access is obviously a great opportunity for HTML page authors. But it also presents a number of challenges when you are designing HTML pages because many people might see your pages on low-resolution TV screens or small hand-held devices. See the "What You Can Do Today to Be Ready for Tomorrow" section, later in this chapter, for some pointers on making sure your HTML pages can be enjoyed and understood by the widest possible audience.

Unity in Diversity

So far in this chapter, I've noted how HTML is in the right place at the right time to enable several key changes in business and interpersonal communication. As the people and companies of the world becomes more interconnected and interdependent, HTML's capability to make all information technology easier to use and less constrained by geography seems almost magical.

Even more magically, HTML has enabled an explosion of new media formats and incompatible file types while at the same time providing the first truly universal format for exchanging all types of information.

But there is no secret mystical force behind this apparent paradox. The power of HTML comes from a very intentional (and seemingly mundane) aspect of its design. Quite simply, HTML standardizes the format of the most common types of information while freely allowing unlimited special cases for proprietary formats and new technology. This means that you can both ensure complete compatibility between the widest variety of software and easily develop unique information formats to meet your individual needs.

Early multimedia Web sites were perhaps the first examples of how to meet these two (apparently conflicting) goals simultaneously. Text and graphics were visible to all visitors (as shown

IPR2024-00143
Apple EX1011 Page 436

in Figure 41.4), while Shockwave movies, Java applets, or other formats were used to provide additional audiovisual or interactive content for those who had the necessary plug-ins or helper applications (as shown in Figure 41.5). Though multimedia formats have come a long way toward standardization, you can still freely include proprietary file formats in ordinary HTML pages and use the helper app, Netscape plug-in, ActiveX object, or Java program of your choice to handle them.

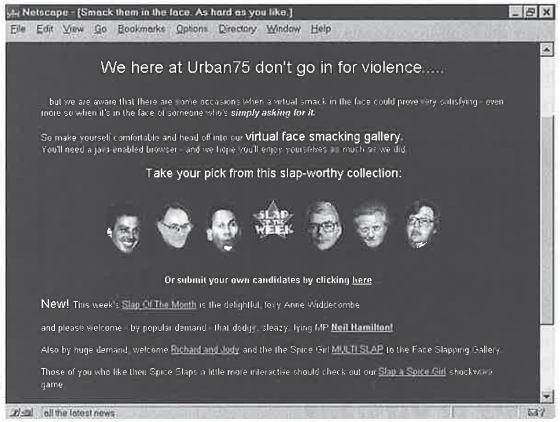

FIGURE 41.4. *This site uses HTML to present text and images that everyone can see.*

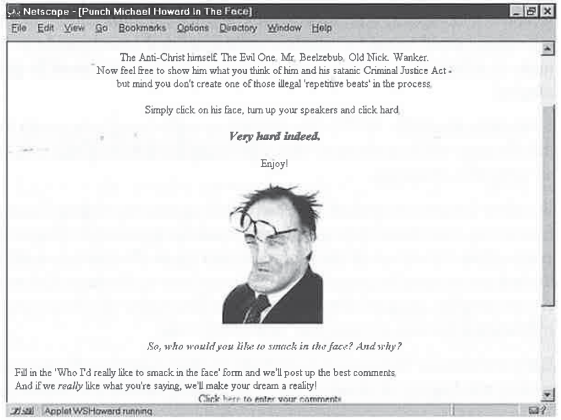

FIGURE 41.5. *If you explore the site shown in Figure 41.4 further, you'll find multimedia elements that require custom applets and/or plug-ins.*

The ability to extend HTML pages with custom data types is far more than a way to embed a nifty movie or virtual reality scene into your Web page. To show how much more, the next two sections highlight some of the most exciting up-and-coming uses of HTML.

HTML as a Programming Language

The near-universal compatibility of HTML provides a big incentive to format any important document as a Web page, even if you have no immediate plans for putting it on the World Wide Web. You can create a single page that can be printed on paper, sent as an e-mail message, displayed during a board meeting presentation, and posted for reference on the company intranet. Or, you can take the traditional route and format the page separately for each of these applications—and edit each file with a different software program when the information needs to be updated. Now that most business software supports the HTML standard, many organizations are trying to get employees to consistently use it for all important documents.

But the great migration to HTML goes beyond what you might have thought of as "documents" in the old days. Combined with Java, ActiveX and other new technologies, HTML is being used for creating full-blown software applications that would traditionally have been written in more unwieldy languages such as C++.

As one example, consider M. Casco Associates, a small technical software company based in rural Pennsylvania. For years, the company has developed and published a line of interactive science simulation software for the educational market using the C and C++ languages. Figure 41.6 shows the Windows version of one of their programs, called Order: A Closer Look at Chaos. (The program uses interactive simulations to teach the basics of chaos theory.)

Because they are a small company, Casco could not afford to develop for more than one operating system or computer type at a time, even though their potential market included DOS, Windows, Macintosh, and UNIX users. Creating an effective interface to combine the tutorial and interactive elements of their software was a significant part of the development effort.

Figure 41.7 shows the latest version of their software, which was rewritten as interactive Java applets embedded in an HTML tutorial and guide. Not only does this new version eliminate much of the interface programming, but it is also instantly compatible with any type of computer that can run a Java-enabled Web browser. As a bonus, M. Casco was able to put part of the program directly on the World Wide Web for potential customers to evaluate without rewriting a line of code. (See for yourself at `http://www.mcasco.com`.) The small Java modules and HTML tutorial are also considerably easier to debug and maintain than a large C++ program requiring a recompile every time a change is made.

Transferring traditional programs to HTML-based presentations usually involves a shift in how you think about the user's experience. Essentially, instead of an application with a hypertext help system, you have a hypertext help system with an application inside it. This gives new users an obvious and friendly way to get to know a program, and still allows the HTML to step out of the way whenever the Java applets or other interactive media should be the focus of the user's attention.

IPR2024-00143
Apple EX1011 Page 437

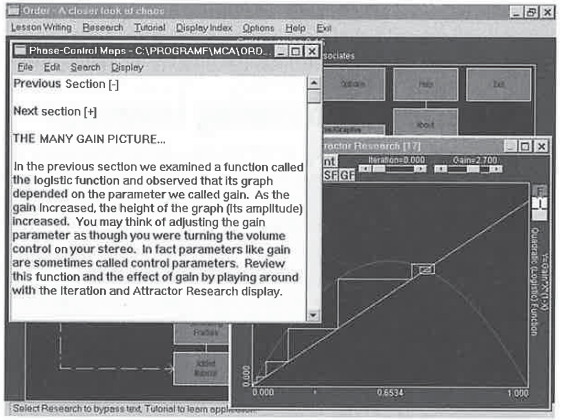

FIGURE 41.6. *This fairly sophisticated Windows program includes both interactive scientific simulations and an extensive online help tutorial.*

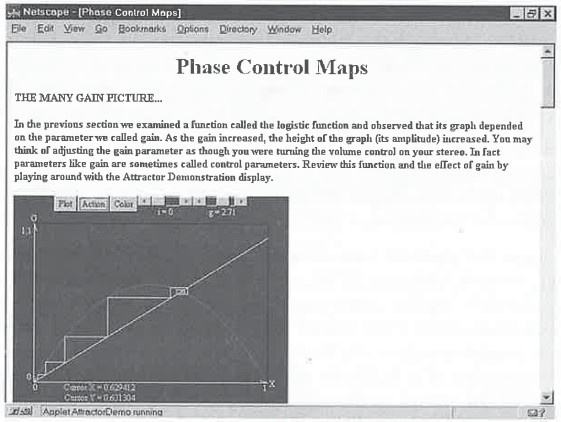

FIGURE 41.7. *With HTML and Java, all the functionality of the large C++ program in Figure 41.6 can be implemented in simple, cross-platform compatible parts.*

HTML Applications of the Future

You've seen that HTML-based presentations can in many cases replace what was once done with proprietary data formats, specialized software, or more traditional programming languages. Here are a few of the other areas where HTML is finding application beyond the Web:

- Kiosks with HTML-based interactive content are popping up everywhere. They look like ATM machines on steroids, and they're helping sell records and theme park tickets, expand department store displays, and even automate the paying of parking tickets. The number of kiosks using intranet and/or HTML technology is projected to soar from under 90,000 today to more than 500,000 by the end of 1998.
- Information-rich CD-ROM titles are migrating to HTML fast. Encyclopaedia Britannica is already entirely HTML based, which enables them to offer their content on CD-ROM, the Web, or a combination of both for maximum speed and up-to-the-minute currency. Because CD-ROM drives display multimedia so much faster than most Internet connections, dynamic HTML presentations become possible that just couldn't be done on today's World Wide Web. The new DVD-ROM drives will be even faster and will hold much more information, making them ideally suited to large multimedia "sites."
- Corporate newsletters are now often created in HTML for the company intranet, and then printed on paper for delivery to employees or customers who wouldn't see them on the Web. The traditional difference between online and paper presentations was that graphics needed to be high-resolution black-and-white for printing and low-resolution color for computer screens. Today's inexpensive color printers, however, do a great job making low-res color images look great in an HTML-based newsletter.
- Teachers are finding that tests and educational worksheets are easier to administer as HTML pages and can include many types of interactive content that wouldn't be possible on paper. Even for students who lack access to the Internet, simple HTML documents can be passed out on floppy disks.
- Vertical market users often buy a computer specifically to run a certain custom-designed application or set of applications. The VARs and systems integrators that provide these systems are delivering machines configured to start up displaying HTML pages. This helps step users through the use of the machine, or replace old-fashioned "idiot menus" with a more attractive and sophisticated interface without sacrificing ease of use.
- Even game developers are getting into the HTML act. While some programmers still contend that HTML-based content will never be the best solution for calculation-intensive applications, others have proceeded to produce Java and ActiveX modules that do high-speed 3D graphics rendering (which is arguably the most intensive application of all) quite nicely. These engines are being used for both Web-based *virtual reality* worlds and cutting-edge entertainment titles to be distributed on CD-ROM.

I could list many more creative and beneficial uses of HTML beyond run-of-the-mill Web pages, but the point should be clear: If you need to present any type of information, seriously consider HTML as an alternative to the software or programming tools that you would have used a couple years ago for the job.

IPR2024-00143
Apple EX1011 Page 438

What You Can Do Today to Be Ready for Tomorrow

If you've made your way through most of the chapters of this book, you already have one of the most important ingredients for future success in the new digital world: a solid working knowledge of HTML.

Chances are that your primary reason for learning HTML at this time is to create some Web pages, but I hope this chapter has convinced you that you'll be using HTML for far more than that in the future. Here are some of the factors your should consider when planning and building your Web site today, so that it will also serve you well tomorrow:

- The multimedia and interactive portions of your site are likely to need more revisions to keep up with current technology than the text and graphics portions. When possible, keep the more cutting-edge parts of your site separate and take especially good care to document them well with the `<COMMENT>` tag.
- Though new technologies such as Java and Shockwave may be the wave of the future, avoid them today except when developing for disk-based media or a fast local intranet. Even when everyone is using the new 33Kbps and 56Kbps modems, many people still will move on to a different site before they'll wait for an applet or interactive movie to download, initialize, and start working.
- Because style sheets will soon give you complete control over the choice and measurements of type on your Web pages, it would be a good idea to study basic typography now if you aren't familiar with it. Understanding and working with things such as *leading, kerning, em spaces,* and *drop caps* have long been essential for producing truly professional quality paper pages. It will soon be essential for producing outstanding Web pages, too.
- One of the most popular and important features that will be added to many Web sites in the near future is interactive discussions and work groups. If you only have time to evaluate one new technology, that might be the one to pick. The new Netscape Communicator 4.0 package has especially strong support for group collaboration and communication.
- When you design your pages, don't assume that everyone who sees them will be using a computer. Televisions, video-telephones, game consoles, and many other devices may have access to them as well. Some of these devices have very low resolution screens (with as few as 320×200 pixels). Although it's difficult to design a Web page to look good at that resolution, you'll reach the widest possible audience if you do.
- Remember that a majority of people are still using versions 2.0 or 3.0 of Netscape Navigator and Microsoft Internet Explorer to view Web pages. If you use HTML 4.0 features such as layering and style sheets, make sure your pages still look good in browsers that don't support those features!

- Whenever you run into something that you'd like to do on a Web page, but can't with HTML as it stands today, include a comment in the page so you can add that feature when it becomes possible in the future.

Summary

This chapter provides a birds-eye view of the future of HTML and the new roles that HTML will play in global communications. It also outlines several applications of HTML that are likely to be much more common in the near future than they are today. Finally, it offers some advice for planning and constructing HTML pages today that will continue to serve you well into the future.

IPR2024-00143
Apple EX1011 Page 439

GLOSSARY

Glossary

by Dennis Báthory Kitsz

IPR2024-00143
Apple EX1011 Page 440

A

ActiveX: A technique developed by Microsoft to run applications on client Microsoft-based computers through an active Web page interface. Like plug-ins, once loaded, ActiveX remains part of the browser. *See also* **helper application, Java, plug-in, viewer**

agent: (*also called* autonomous agent, information agent, intelligent agent) An application or applet that carries the user's preferences into a virtual environment such as the Web and acts independently on those preferences. Infinitely configurable, an agent is the user's doppelgänger in cyberspace.

American National Standards Institute (ANSI): An organization dedicated to promoting national technology standards. ANSI is privately funded and includes both public and private members. *See also* **International Organization for Standardization**

American Standard Code for Information Interchange (ASCII): A code that represents a set of 256 unformatted characters—letters, numbers, symbols, and terminal actions (such as carriage returns and line feeds) in a standard format of 8 bits (although 128-character, 7-bit ASCII exists as well).

anchor: A portion of HTML code that identifies a specific document or a position (destination) inside a specific document. Also, the visible portion of the code, usually represented by underlined words. *See also* **hyperlink**

ANSI: *See* **American National Standards Institute**

applet: A dependent piece of computer software, usually small and compact, that requires an application with or within which to operate.

application: A complete piece of computer software, whether it is a compact utility consisting of a few bytes or a suite of programs that work together. An application stands alone inside its environment processor, operating system, browser, and so on.

ASCII: *See* **American Standard Code for Information Interchange**

attribute: Additional information added to element tags to specify characteristics of or render the element, such as size, color, borders, and other qualifications. Attributes are author-specified values.

authentication: A software technique that guarantees the identity of an individual or company; used in secure sites.

B

bandwidth: The capacity to carry data. In a network, this is usually bits per second and is critical for browsing speed, especially in sites heavy in hypermedia content.

bit: A binary digit, the smallest atom of digital information, represented by the number 0 or 1.

bitmap graphics: The resolution of an image into a series of dots, represented by a map of binary digits (bits). *See also* **raster graphics, vector graphics**

body: The "visible" content of an HTML document that a browser displays.

bookmark: A popular term for a hyperlink name and URL that the user saves for future use in a bookmark list or hotlist.

block element: A tag within an HTML document that identifies the document's structural areas by grouping text content in terms of its meaning. *See also* **container, text element**

bps: Bits per second. *See* **Kbps**

browser: Client software for displaying Web pages and using hyperlinks to navigate the Web.

byte: A combination of eight bits that constitutes a basic digital "word" size, capable of forming 256 possible combinations of bits.

C

C and **C++:** Advanced programming languages that dominate network applications. C is an older, structured language; C++ is its object-oriented successor.

cable-modem: A device that combines television and telephone technology to enable high-speed receipt of information via cable and normal-speed transmission via modem. The cable-modem is anticipated by content providers because it will make Web page (and hypermedia) downloads appear instantaneous while keeping data-thin and slower user interactions on ordinary phone lines. *See also* **modem, push**

cache: A temporary storage area for data. Specifically in terms of browsers, a cache is the trail of Web documents stored on the user's computer so that the user can instantaneously view a document again. Because some Web sites frequently update documents, caching can have the unwelcome side effect of holding on to "stale" pages.

cascading style sheets (CSS): A layout and display system for Web pages. The user can specify fonts and sizes, layouts, colors, and other viewing and printing parameters. Because CSS are optional and not part of HTML itself, they do not break HTML's platform independence for layout purposes. *See also* **frames, layers**

cell: A portion of a table in which information is displayed. A cell can encompass one or more of a table's rows or columns.

CERN: *See* **European Laboratory for Particle Physics**

CGI: *See* **common gateway interface**

character encoding: A numerical range of codes, repertoire of characters, and their mapping. *See also* **character set**

IPR2024-00143
Apple EX1011 Page 441

character set: The letters, numbers, and symbols available for display or printing. Character sets in HTML are defined to provide platform-independent consistency in all display systems for all languages.

chunk: A manageable piece of information. The definition is flexible but usually means the division of larger text into a single theme, concept, topic, or idea. Good "chunking" helps present nonlinear information.

class: The definition for a type of object in the form of a general software blueprint used in object-oriented programming. Classes may share elements with each other and may inherit characteristics of the parent blueprint. *See also* **inheritance**

client: A combination of computer and software capable of receiving information and instructions from a remote host computer or server.

client-side: Applications, activities, or events that take place on the client computer, such as image maps or applets.

code page: The character encoding developed for a specific language. *See also* **character encoding, character set**

common gateway interface (CGI): A programming interface between the client (browser) and server. CGI accepts data from the user and acts on it, sometimes returning information to the user. Common examples are forms and searches.

compression: The reduction of raw data by removing redundancies, making predictions, and describing events, rather than replicating them. This allows for smaller files and faster downloads but, especially in the case of audio, noticeable degradation.

container: An HTML element made up of an opening and closing tag. Information is contained within the tags. *See also* **block element, text element**

content: The information a Web page provides. Content is often highly debated and often redefined to defend a Web page creator's design. Web purists largely view content as text.

cookie: A line of text stored in the client browser that the server can retrieve. It may contain encoded passwords, shopping-cart contents, user identification, and so on. Cookies are controversial because commercial Web designers use them extensively to track visitors.

copyright: An international treaty setting the right to intellectual property. Although copyright exists the moment an idea is manifested as a work of authorship (in any medium), the nature of the Web culture has not taken copyright into account until recently.

Cougar: The latest W3C proposed draft for HTML specifications that distinguish between markup (content) and appearance of Web pages. Like most HTML specifications, it lags behind and attempts to codify existing practice into universally accessible definitions and incorporates significant access techniques for hypermedia. Cougar may become HTML 4.0, succeeding the present 3.2 specification.

CSS: *See* **cascading style sheets**

D

database: An organized collection of information stored on a server.

declaration: A statement about a document and its structure, including its type, syntax, scope, character set, entities, naming rules, and so on.

dialog box: A box displayed on the screen that asks for user selections. The dialog box is a core component of a graphical user interface.

dialup: The method of accessing the Internet via modem and traditional telephone lines.

DNS: *See* **Domain Name Service**

document type definition (DTD): The hierarchy of elements used to define the structure of a document. The DTD is set forth in the HTML specification.

domain name: A name for an organization, business, or government agency expressed in word-like form. There are national domains and six major organization domain types (`.com`, `.edu`, `.org`, `.gov`, `.mil`, and `.net`). Combined with a server name, they are translated by domain name services into a numeric IP address called the "dotted quad" (such as 204.246.11.197).

Domain Name Service (DNS): A computer containing a cross-reference of domain names to respective numerical IP addresses.

DTD: *See* **document type definition**

dynamic: In Web parlance, particularly in terms of Web pages, dynamic refers to server responses created based on active user input (such as forms) or passive user input (such as browser identification or cookies).

dynamic HTML: A method that Microsoft promotes to produce pages with active or changing information.

E

element: Though commonly referred to as a tag, an element is in fact a component of hierarchical structure defined by a document type definition. The element is actually described (or delimited) by the tag. This clarity is obscured, however, by elements without contents (such as line breaks), and those that may not contain other elements (such as anchors). *See also* **block element, text element**

embedded object: An object included as part of a document in its complete form, such as an illustration, audio file, or application. *See* **object**

encoding (character): *See* **character encoding**

encryption: A method of altering information to make it unreadable by anyone except users who possess the rules to decode it. Public-key encryption is popular because anyone can encrypt and send a message, but only the recipient can decode it.

IPR2024-00143
Apple EX1011 Page 442

entity: A character that cannot be typed on the keyboard (such as é or ©) or that has a special meaning to HTML (such as <). It is represented by its numeric form (called a character reference: `©` for ©) or mnemonic form (`©` for ©).

European Laboratory for Particle Physics (CERN): *Conseil Européan pour la Recherche Nucléaire.* This laboratory in Geneva was the birthplace of the Web concept and execution in 1989.

event: An incident or occurrence meaningful to a program, such as an operation's completion, a request for data, or an interruption of activity.

event handler: A portion of a program that responds to an event by processing its details, such as a mouse click, keyboard stroke, or "out of paper" signal.

Extensible Markup Language (XML): A subset of SGML that parses a document's structure and elements in terms of its document type definition.

extension: (1) The method of identifying the purpose of files and applications within the filename, following a dot. Common to the Web as well as UNIX and DOS operating systems, extensions can identify files such as HTML (`index.html`), MIDI (`song.mid`), images (`picture.jpg`), and so on. File extensions are not found in some operating systems, such as Macintosh.

extension: (2) Additions to HTML outside the existing specifications. Companies often contribute, encourage, promote, and even implement these extensions to advance features they would like to include in future releases of the HTML specification.

extranet: Internal web sites (intranet sites) accessible to the outside world, usually via passwords. *See also* **intranet**

e-zine: An online electronic publication, specifically (like a print 'zine) an amateur, artistic, or non-mainstream publication and presentation.

F

fat: In reference to a client, *fat* is a general term indicating a machine that is a complete, standalone computer with an extensive operating system and powerful applications. Although such a computer can access a server, it does not require one.

file path: *See* **path**

file transfer protocol (ftp): The method for remote login to a computer to exchange files over a TCP/IP network. Ftp usually requires a username and password, but anonymous ftp can also be made available, especially with Web browsers.

firewall: Protective software that prevents unauthorized users or software from entering an internal network. It is used to protect theft of data and to keep out viruses but has the side effect of making Web use difficult, particularly for hypermedia.

font: A set of typographical characters (numbers, letters, and symbols). A complete Western (Latin) font, for example, includes upper- and lowercase letters as well as a full range of accented characters. Special software fonts such as Cherokee have helped sustain and expand their respective cultures. *See also* **kerning**, **monospaced font**, **proportional font**

form: An HTML component that enables Web authors to have input fields on their pages, permitting feedback from users and offering interactive options.

format: The presentation of a document, including fonts, voice, paging, colors, and other features. Format is also achieved by usage (such as paragraphs, lists, and titles) and logical elements. *Compare* **cascading style sheets**, **layout**, **structure**

frames: Valuable but confounding extensions to HTML that present information as a concurrently displayed group of pages, each with a unique URL but concatenated by a master document. Frames are a kind of simultaneous hypertext. Because they are visually organized, they are not considered platform-independent and are the subject of bitter controversy. *See also* **cascading style sheets**, **layers**

ftp: *See* **file transfer protocol**

G

GIF: *See* **Graphics Interchange Format**

glyph: A character in an ideographic writing system, usually not phonetically based. Ideographic systems can use tens of thousands of unique glyphs, as opposed to the hundred or so in phonetic systems.

gopher: An early method of accessing information on the Internet consisting of document indexes, file transfer, database browsing, and Telnet.

graphical user interface (GUI): Generally, an operating environment in which icons represent files and applications on a monitor display, and a point-and-click device (such as a mouse, trackball, or pen) is used to activate them. GUIs usually employ drag-and-drop file movement and launching applications, plus buttons, toolbars, drop-down menus, and other conventions not available with line-based (keyboard) methods.

Graphics Interchange Format (GIF): An image compression method CompuServe developed and the first that Web browsers accommodated. Files are large and the palette is limited to 256 colors, but no information is lost in the compression.

GUI: *See* **graphical user interface**

IPR2024-00143
Apple EX1011 Page 443

H

handheld personal computer (HPC): A complete computer in a small package, with a limited display. The HPC and its kin concern Web page authors because the small display and limited features call for HTML's platform independence.

head: Part of the HTML document containing information about the document itself, including the document title, searchability, keywords, and other contents including scripts.

header: The client sends the HTTP request header to the server to indicate what information it wants. Among those requests are document title, body, authorizations, contents, MIME types, and interaction with CGI.

helper application: A program that is not a native part of a Web browser nor a software plug-in provided for the browser. Helper applications are used to display non-text content, particularly newly developed or custom multimedia content, that goes beyond the browser's capabilities at the time of its release. *See also* **ActiveX, Java, plug-in, viewer**

hex (hexadecimal) notation: An effective method of representing binary numbers in systems with a 4-bit architectural heritage. Four binary digits (0000 through 1111) can be represented by hexadecimal numbers 0 through F; eight digits, 00 through FF; sixteen digits, 0000 through FFFF, and so on.

hit: A visit to a Web page or, sometimes, to each file on that page.

host: A server. In Web terms, a computer with a specific name on the Internet. *See also* **node**

hotspot: The portion of an image that is mapped to a hyperlink.

HPC: *See* **handheld personal computer**

HTML: *See* **HyperText Markup Language**

HTTP: *See* **HyperText Transfer Protocol**

HyperCard: The first practical implementation of hypertext theory in the Apple computer.

hyperlink: A connection between one file and another file or inside a file. It is one of the HTML anchors.

hypermedia: Multimedia as presented on the Web, including passive and active graphics, audio, video, virtual reality, and other media concepts under development.

HyperTalk: The scripting language used to create Apple HyperCard documents. *See also* **HyperCard, script**

hypertext: The practical and conceptual heart of the Web, hypertext is a system of relating points within and outside a text to each other nonlinearly. Hypertext, as manifested with hyperlinks, moves a user from text to text at will.

HyperText Markup Language (HTML): A platform-independent method of identifying a document's structure and references. In its ideal form, HTML makes no reference to the ultimate user's hardware; in practice, HTML includes presentation information as well as structural elements.

HyperText Transfer Protocol (HTTP): A protocol for transferring data formats between a server and client. Data formats include plain text, hypertext, images, sound, public or proprietary formats specified as a MIME type, and metainformation about the data.

I

icon: A visual representation of a file or application.

image (IMG): Originally created to include visual images on a Web page, the IMG tag has expanded to include many types of hypermedia, including sound, video, and interactivity.

image map (imagemap): An image with hotspots that serve as mouse-activated hyperlinks.

inheritance: Characteristics of a class shared with a parent or child. *See* **class**

integrated services digital network (ISDN): Digital communications standards for voice, data, and video over standard phone lines at 128 Kbps. It is better established in Germany and is developing slowly in the U.S. because of anticipation of cable-modem services.

intellectual property: The ownership of an idea's manifestation—that is, a work put into communicable form. The communicable form can include paper, electronic media, or stone, and encompasses all forms of authorship. It has been the subject of debate and revisions to the copyright treaties because of the "Wild West" nature of the Web. *See also* **copyright, watermark**

interface: A generic term for the method used to join hardware components, software components, or users and systems. For example, a user interface is a concept and method of interaction; a printer interface consists of cabling, electronic components, and software drivers; a data interface is software to translate data formats; an equestrian interface is a saddle and reins.

International Organization for Standardization (ISO): Founded in 1946, this voluntary, nontreaty organization creates international standards and definitions. These standards apply to both hard and soft technologies, methods, and contents, and they are not limited to any particular group of fields or subjects. *See also* **American National Standards Institute**

Internet: A worldwide network of computers in communication using agreed-on protocols. A simple definition does not explain the Internet as a *concept*, but it is an opportunity for universal access to human knowledge, wherever it is distributed. *See also* **nonlinear, World Wide Web**

Internet Explorer: A Microsoft Web browser that gained popularity by being bundled with the Windows 95 and NT 4.0 operating systems.

IPR2024-00143
Apple EX1011 Page 444

Internet Protocol (IP): A system of network connections using numerical (IP) addresses. No direct connection between the sending and receiving computer is assumed; the IP address is used to establish an indirect pathway through multiple networks.

Internet service provider (ISP): A business that offers connection to the Internet, usually through dialup or dedicated telephone lines. ISPs can be little more than a local server, a full-featured system of various accounts including server Web space, or a commercial provider with proprietary features and chatrooms along with Internet access.

intranet: A network internal to an organization or corporation set up to mimic Internet features. Intranets may also connect to the Internet.

IP: *See* **Internet Protocol**

IP address: A quartet of digits separated by dots (such as `204.246.11.197`) that represents a unique computer connected to the Internet. Domain names are resolved to IP addresses before information is exchanged.

ISDN: *See* **integrated services digital network**

ISO: *See* **International Organization for Standardization.** (ISO is actually not an acronym, but the prefix for the standards, derived from the Greek prefix *iso*, meaning "same.")

ISP: *See* **Internet service provider**

J

Java: A programming language Sun Microsystems developed to permit standalone applications to run on a client computer through an active Web page interface on a "Java virtual machine." *See also* **ActiveX, helper application, plug-in, viewer**

JavaScript: A scripting language that is interpreted and runs on a client computer within an HTML document and provides active and interactive applets to Web pages.

JavaScript style sheets: A page formatting method that Netscape promotes and JavaScript controls. *See also* **cascading style sheets**

Joint Photographic Experts Group Format (JPEG): A method of compressing images; the second such method adopted to display images on the Web. The palette is large (16 million colors) and file sizes can be made small with JPEG's variable compression, but the method is *lossy*, meaning that the original image's detail is not fully restored during decompression. It is used largely for photographs, where the eye makes up for compression losses.

JPEG *See* **Joint Photographic Experts Group Format**

K

Kbps: A thousand bits per second, often used in reference to Internet connection speeds. Early modem speeds were 300 bps (bits per second), whereas 28.8 Kbps is common today, with compression that squeezes an effective rate of 34 Kbps through the line. Other frequently encountered speeds are ISDN at 128 Kbps, T1 connections at 1.5 Mbps (million bits per second), T2 at 3 Mbps, and T3 at 45 Mbps. *See also* **bps, Mbps**

kerning: A typographical technique that overlaps characters for visual balance.

L

layers: A Netscape-specific method of organizing multiple documents on a page, including layout features and identification of viewable content. Layers are related to frames but offer more author control. *See also* **cascading style sheets, frames**

layout: The visual appearance of a document. Because HTML identifies a document's structure rather than its appearance, Web authors have sought techniques to make their pages display to their tastes. Although some HTML workarounds have been developed, along with display elements within the HTML specification itself, advanced layout calls for other methods, including images and cascading style sheets. *Compare* **cascading style sheets, format, structure**

LCD: *See* **liquid crystal display**

leading: The vertical space between lines of text.

linear: A concept of organizing information in a known sequence. With few exceptions and until recently, this has been the method of storing, retrieving, and exchanging information throughout the history of the written word. Linear information is considered "self-contextualizing" because the references appear in a past, present, and future sequence.

link: *See* **hyperlink**

Linux: An operating system based on UNIX and created as freeware. Many small Internet servers run Linux, which independent programmers continue to maintain.

liquid crystal display (LCD): Low-power display hardware in which a liquid becomes semi-opaque in the presence of an electrical current.

listserv: An automated mailing list for exchanging messages. Documents sent to a listserv are redistributed to all its subscribers.

LiveScript: *See* **JavaScript**

logical markup: The identification of a text's characteristics, content, and structure. *Compare* **physical markup**

IPR2024-00143
Apple EX1011 Page 445

Lynx: A text-only browser found on older or slower computers, sometimes used for high-speed surfing or text-to-speech access. It has a hardcore following and is also useful for testing both the platform independence and the potential ambiguity (or complete opaqueness) of a Web page design.

M

map: *See* **image map**

markup: The identification of a text's characteristics, content, structure, and presentation by use of language-like descriptions. *See also* **logical markup, physical markup**

Mbps: Million bits per second. *See* **Kbps**

menu: A list of choices. On Web pages, menus can be presented as hyperlinks, drop lists, check-box groups, maps, and even hypermedia controls.

MIDI: *See* **Musical Instrument Digital Interface**

MIME: *See* **Multipurpose Internet Mail Extensions**

modem: *Modulator-demodulator.* A device that converts digital information into a sound format that can be transmitted over acoustic phone lines.

monospaced font: Characters displayed or printed in equally spaced "typewriter" format.

Moore's Law: Originally put forward in 1965 by Intel co-founder Gordon Moore, it states that computer power doubles every 18 months. It has held true for 30 years. Software and hardware creators have exploited technological developments to the chagrin of budget-conscious users and have presented a formidable barrier to those concerned with universal access to computing power.

Mosaic: The first successful graphical browser, created in 1993 at the NCSA.

Motion Picture Experts Group (MPEG): Video and audio compression standards. There are presently three MPEG standards with several "layers" each.

Mozilla: From "Mosaic killer," the original prototype name for Netscape Navigator.

MPEG: *See* **Motion Picture Experts Group**

multimedia: A term expropriated from the art world (where it means a performance or artwork including several human senses), multimedia is commonly used to mean the integration of at least audio and video elements into a computer presentation. Its Net equivalent (*See* **hypermedia**) is the bane of pure HTML advocates because it does not offer platform independence.

Multipurpose Internet Mail Extensions (MIME): A method of encoding binary data for electronic mail transmission, including audio, graphics, video, word processing, music scores, and applications. MIME was also adopted as the standard for Web transmission.

Musical Instrument Digital Interface (MIDI *or* **Midi):** The specification of sonic parameters (such as note, volume, duration, and controllers) as a serial stream of data and the specification of the required connectivity. MIDI stepped out of the production studio onto the Web because of both the compact size of the files and its platform independence.

N

National Center for Supercomputing Applications (NCSA): The University of Illinois at Urbana-Champaign center responsible for developing the original Mosaic browser.

navigate: To move through the Web using hyperlinks. Also, to move through Web pages using hyperlinks, droplists, toolbars, maps, hypermedia controls, and other techniques.

NC: *See* **Network Computer**

NCSA: *See* **National Center for Supercomputing Applications**

Netscape Navigator: Web browser created by NCSA expatriates (*See* **Mozilla**), quickly rising to popularity by a combination of aggressive giveaway marketing and the promotion of presentation-oriented Web pages, graphics, sound, and other hypermedia and a plug-in structure to accommodate new forms of media.

Network Computer (NC): A simplified computer (sometimes called a *thin client*) whose applications and files are stored on one or more servers, rather than locally, and without the features of a full operating system.

news: *See* **Usenet**

node: (1) The point of entrance to or exit from a Web page, considered a standalone construct.

node: (2) A computer connected to a network capable of generating, receiving, or forwarding data.

nonlinear: A concept of storing, obtaining, and exchanging information through multiple, indirect, layered, and sometimes unpredictable paths with no predefined order. Human thinking frequently follows such paths, but until modern technology made it possible, media representation of nonlinear information was virtually unknown outside its rudimentary use in printed footnotes or operatic motifs. It is now exemplified by the Web's hyperlinks.

IPR2024-00143
Apple EX1011 Page 446

O

object: A self-contained piece of information ranging from a single byte to a complete document or application.

object-oriented: Programming and databases using self-contained and interchangeable software routines or data elements rather than linear (and often custom) coding. Focusing on ways of interacting with data, object orientation is based on building-block thinking.

ODBC: *See* **Open Database Connectivity**

Open Database Connectivity (ODBC): Microsoft standard for exchanging software components and data within applications.

operating system (OS): The software control system for a computer. It usually refers to a disk-based system, although single-board and harsh-environment computers may use ROM-based operating systems. The operating system may have several layers to create an environment (such as older Windows operating through DOS) or work directly with the computer (such as the Macintosh, Windows 95, or UNIX).

OS: *See* **operating system**

P

page: A Web document in HTML with a single URL. Pages are usually *chunks* (*See* **chunk**), but in the case of linear information, pages can sometimes be long, text-like documents or lists of links.

parameter: A range of possibilities or actions in an application.

parse: To examine code or markup for underlying structure, grammar, and syntax; to determine whether errors are present; and to prepare the results for an action—page presentation in the case of HTML.

password: String of characters known only to a system or application and its user. Passwords are a simple method of keeping information confidential, but their proliferation has complicated lives for frequent computer users. Authentication methods have been proposed to replace passwords in some situations.

path: The trail through a computer's directory structure to a document.

PDA: *See* **personal digital assistant**

Perl: *See* **Practical Extraction and Report Language**

personal digital assistant (PDA): A small, handheld, computer device with a subset of the features found in larger operating systems. PDAs with Web-browsing capabilities present a challenge for page designers and HTML authors because of their small monochrome displays.

physical markup: The markup of a document's presentation. Physical markup is consistent with standard HTML practice in which it contributes to structure and meaning as identified in the typographical world. This remains a contentious issue, as in the case of emphasized and strong text versus italic and bold text. Some recognize the former as creating well-formed HTML with a structural meaning; others recognize the latter as creating well-formed HTML with typographical meaning.

PICS: *See* **Platform for Internet Content Selection**

pixel: From *picture element*, the pixel is the basic element on a raster display (*See* **raster graphics**), represented as a dot containing intensity and color information.

platform: The essential combination of hardware (microprocessor, architecture, and memory organization) and software (operating system and applications) that makes up a computer. Because hundreds of platforms exist beyond the well-known DOS, Windows, Macintosh, and UNIX systems, the HTML author strives for platform independence.

Platform for Internet Content Selection (PICS): Proposed standard that gives parents the technical means to block indecent material from children. It defines file formats for independent content rating, and the published results can be used by PICS-enabled software.

platform independence: The capability of applications or data to be used on any computer. Never actually achieved, platform independence is attempted by HTML by identifying a document's structure and liberating a document's elements from their presentation.

plug-in: A dependent application that extends a browser's capability to present advanced or custom content without changing the browser itself. Originally developed as helper applications, plug-ins are now provided by media content developers for the most popular browsers. *See also* **ActiveX, helper application, Java, viewer**

PNG: *See* **Portable Network Graphic**

port: A hardware or software entry point to a computer system, largely software ports in Internet protocol terms (HTTP uses port 80, Telnet port 23, IRC port 6667, and so on).

Portable Network Graphic (PNG): Compressed graphics format with a large color palette and low information loss. PNG is a public-domain technique that has been proposed as a replacement for GIF. *See also* **Graphics Interchange Format**

Practical Extraction and Report Language (Perl): The most popular language for programming CGI. *See* **common gateway interface**

proportional font: Characters presented according to their width rather than equally, as exemplified by handwriting and typesetting. Proportional fonts have become the graphics interface standard, but monospaced fonts are still used in terminals and for special circumstances.

IPR2024-00143
Apple EX1011 Page 447

protocol: A standard for network communication. Different protocols exist for unique purposes: HTTP moves Web documents, FTP transfers files, TCP manages network traffic, IP handles data packets. A group of simultaneously operating protocols is a stack. Protocols are also used for communications with peripheral devices such as printers.

proxy gateway: Intermediary computer and software that pass data between an isolated computer and the outside world. Proxy gateways can be configured to pass only "safe" information to the client. *See also* **firewall**

proxy server: A computer that stores and distributes documents it has obtained from other servers. This technique is used to lower access costs or keep frequently used documents at hand. One disadvantage is that updated Web pages might not be available to clients until the proxy server is next updated. *See also* **cache**

pull: Actions initiated by a client. Most browser applications request, or *pull*, information from the server.

push: Actions initiated by a server, especially for sending updated information to a client browser, such as news, stock quotes, and live video.

Q

QuickTime: Proprietary Apple format for audio and video frequently found in CD-ROM applications and occasionally found on the Web. *See also* **MPEG**

R

RAM: *Random-access memory.* Advanced clients require large amounts of high-speed RAM to handle graphics, sound, and video without requiring frequent access to a hard disk.

raster graphics: A method of drawing images by altering the intensity and color of an electron beam sweeping across video display terminal phosphors in a regular pattern of horizontal lines—the raster. It is an established television-style technology.

RealAudio: A popular audio compression system capable of streaming audio on the Internet. RealAudio offers free encoders and décoders but charges for the live-streaming servers. RealAudio automatically detects a computer's Internet connection speed and configures itself for highest audio quality.

Request for Comments (RFC): Published documents that include approved and proposed Internet standards, as well as agreed-upon plans and concepts. They are independently proposed by technical experts and remain RFCs even if they become standards.

RFC: *See* **Request for Comments**

robot: Automated software that performs tasks independent of users, often said to "live online." Also called a bot or an agent. *See also* **agent**

S

script: A written set of instructions for performing tasks, a script is usually application-specific commands that customize an application's operations. Scripts are found in most Internet applications, with examples found in the Web's JavaScript, in dialer scripts, IRC scripts, server scripts, and so on.

scrolling: The movement of screen information horizontally and vertically. Scrolling is important to Web authors both as a convenience issue (vertical scrolling is expected, horizontal scrolling is considered inconvenient) or access issue (sometimes, as with WebTV, horizontal scrolling is impossible).

search: To use Web-based, interactive tools to locate specific information or Web sites; it is the opposite of surfing.

Secured HyperText Transfer Protocol (SHTTP): An HTTP protocol in which an encrypted connection is established before data is exchanged.

security: The preservation of privacy in data exchanges. Security usually calls for encrypting data and authenticating users and may also include passwords.

Sequential Query Language (SQL): Database-access and database-retrieval language used for searching large quantities of information and answering questions; pronounced "sequel." Other popular database techniques are Informix, Oracle, and Sybase.

server: A combination of computer, software, and network connection that acts as a central location and acts on requests for resources, such as applications, Web pages, or other data. It also processes responses to interactive pages, such as forms or searchers.

server-side: Applications, activities, or events that take place on the host computer or server, where information and responses are processed.

SGML: *See* **Standard Generalized Markup Language**

ShockWave: A popular browser plug-in with which to create animation and sound and, recently, other hypermedia content. Newer versions of ShockWave use streaming techniques. *See also* **streaming**

shopping cart: A software method to keep track of online purchases. Shopping carts are difficult to implement because the server cannot know the browser's location on the Web. Web authors resort to cookies, scripts, and other techniques to maintain shopping carts.

SHTTP: *See* **Secured HyperText Transfer Protocol**

site: *See* **Web site**

socket: A method for making a "virtual connection" between software processes by identifying data type, direction, and port. Originally developed for UNIX, it is now used by other systems including Windows and Macintosh.

IPR2024-00143
Apple EX1011 Page 448

spider: A searching robot that explores and indexes the contents of all local hyperlinks from a given Web page.

SQL: *See* **Sequential Query Language**

Standard Generalized Markup Language (SGML): An international standard for text information processing, distribution, search, and retrieval. SGML documents have content and structural information about content. Format is not specified in SGML.

stream: A continuous flow of information.

streaming: The use of a data stream to present time-based content (such as audio or video) as it happens, without waiting for a complete file to download. Streaming can be used with existing complete documents or "live" from digital information created on the fly.

structure: The organization of a document. In HTML terms, structure identifies the markup of consistent elements that identify their purpose; the definition of its content. Structuring a document frees the viewer to determine its use, particularly in a nonlinear, hypertext situation. *Compare* **format, layout**

style sheets: *See* **cascading style sheets**

surf: To browse or navigate Web sites without a specific informational goal in mind. *Compare* **search**

T

table: Originally conceived as a way of marking up tabular information, the table has evolved into a document presentation element. Using borderless tables, Web designers achieve layout-like results in graphical browsers. The use of tables for design is controversial because they are used to format a display rather than to provide document structure.

tag: Markup instruction in an HTML document presented in angled brackets. A tag may contain a declaration, statement, and so on, as well as attributes.

TCP: *See* **Transport Control Protocol**

TCP/IP: *See* **Transmission Control Protocol/Internet Protocol**

Telnet: Remote login to a computer that enables a user to use a computer from a distance, including manipulating files and executing applications.

template: A document or file containing generic information used to build new or updated documents by "filling in the blanks" or altering variable information. Templates are particularly useful for Web sites that update information frequently, maintain an identical presentation and structure, and archive it all (such as online publications, reviews, and directories).

terminal: The interactive user end of a network, usually only a display, keyboard, and interface, but sometimes generally referring to any interactive end-user point on a network, including full-blown computers or, in earlier days, Teletype machines.

text element: A tag within an HTML document that identifies the characteristics of words or characters, rather than the structure of the content. *See also* **block element, container**

thin: In reference to a client, *thin* is a general term that indicates a machine without the computing capability of a complete standalone computer with an extensive operating system and powerful applications. Most of its power comes from a server.

toolbar: (*sometimes* **navigation bar**) A set of icons that are hyperlinks to other Web pages or hypermedia.

Transport Control Protocol (TCP): A means by which a connection is established between a client and a server. Data is broken into packets that are addressed to the receiving computer together with information for their reassembly.

Transmission Control Protocol/Internet Protocol (TCP/IP): The combination of client/server connection and data transfer (TCP) and network connection system (IP) used on the Internet. *See* **Transport Control Protocol, Internet Protocol**

transportability: *See* **platform independence**

U

UCS: *See* **Universal Multiple-Octet Coded Character Set**

UCS Transformation Format: A method of switching between standard ASCII and UCS character sets within a serial stream of data.

Unicode: 16-bit character encoding system designed to display phonetic and ideographic languages. Although incomplete, Unicode is implemented on the latest generation of Web browsers.

Uniform Resource Citation (URC): A set of attributes that describe an object's characteristics, such as size, origin, authorship, copyright, data type, and so on.

Uniform Resource Identifier (URI): Generic set of identifications that encompasses URCs, URLs, and URNs.

Uniform Resource Locator (URL): The path to a specific document on the Internet that consists of the protocol (`http://`, `ftp://`, `gopher://`, `news://`, and so on), a domain name or IP address, a directory structure, and the document name.

Uniform Resource Name (URN): Any identifier other than a URL. This seemingly ambiguous definition leaves open the door to flexible ways of accessing Internet information in the future.

IPR2024-00143
Apple EX1011 Page 449

Universal Multiple-Octet Coded Character Set (UCS): A 32-bit character encoding system designed to accommodate phonetic and ideographic (glyph) writing systems.

UNIX, or Unix: The world's dominant operating system, dating from the early 1970s. UNIX is considered to provide the optimum combination of portability, flexibility, and power.

URC: *See* **Uniform Resource Characteristic**

URI: *See* **Uniform Resource Identifier**

URL: *See* **Uniform Resource Locator**

URN: *See* **Uniform Resource Name**

Usenet: Growing out of local electronic bulletin boards, Usenet is an informal collection of tens of thousands of thematic groups. Users post ASCII or binary messages that propagate from server to server, sometimes over a period of days. Usenet traffic ranges from authoritative to fictional, accurate to fraudulent, art to pornography.

username: A computer user's identification, nickname, or "handle" for a specific site, application, or connection. For security, the username is combined with a password or other authentication method.

UTF: *See* **UCS Transformation Format**

V

valid document: Usually referring to a Web page, a valid document passes all the requirements of the particular HTML specification under which is was created, including structure, syntax, and characters.

Validator: An application that checks a document for validity.

value: A numeric or text variable that determines the range or function of a parameter. *See also* **parameter**

VBScript: Microsoft entry into the scripting languages for Web sites. *See also* **JavaScript, script**

vector graphics: A technique of displaying images by drawing them directly with lines and curves of changing intensity and color. *See also* **bitmap graphics, raster graphics**

viewer: A helper application originally intended for graphics, then audio. Largely superseded by the term *helper application. See also* **ActiveX, helper application, Java, plug-in**

Viola: An early UNIX-based browser organized on the HyperCard model.

Virtual Reality Modeling Language (VRML): A method of displaying, moving within, changing, and hyperlinking three-dimensional scenes. VRML deals with concepts and techniques similar to HTML but is concerned with the unique non-textual aspects of three-dimensional images, such as lighting and shadow, viewpoints, backgrounds, and the z-axis.

virus: A program, usually destructive or mischievous, designed to hide inside digital information, usually an application or active library, and remain outside the control of the user. Viruses usually self-replicate, hence the name, to infect other programs or systems via diskettes or networks.

VRML: *See* **Virtual Reality Modeling Language**

W

W3C: *See* **World Wide Web Consortium**

WAIS: *See* **wide area information service**

watermark: A technique that permanently marks a digital document, usually with its creator's name and rights information. Digital watermarking is at its infant stage because of the public nature of Web documents. *See also* **copyright, intellectual property**

Web: *See* **World Wide Web**

Web page: *See* **page**

Web site: A combination of related Web pages gathered under one domain or directory, usually dealing with a single theme or personality.

WebTV: A television-based Web appliance, usually containing a subset of Web features, operated by remote control. WebTV concerns Web designers because of its smaller display size and lack of horizontal scrolling.

well-formed document: A document that not only passes HTML markup validation but also possesses a logical structure and is platform-independent.

wide area information service (WAIS): An early Internet method of indexing and retrieving information.

window: A portion of a display screen that contains an operating application or open file. Windows are often thought of as the pieces of paper on a desk (hence the term *desktop*).

Windows: A Microsoft operating system. Windows 3.*x* was a shell formed over DOS, whereas Windows 95 is a true operating system that also contains a Web browser and dialup software. Windows NT is a network system that is operating an increasing number of Web servers.

World Wide Web (WWW): The Web is a subset of the Internet that implements the concept of hyperlinks to exchange information and operates according to the HyperText Transfer Protocol. The Web is distinguished as the first widespread implementation of nonlinear information concepts.

World Wide Web Consortium (W3C): An industry consortium to develop common standards for the Web. Members propose standards and protocols for use of hypermedia.

IPR2024-00143
Apple EX1011 Page 450

WWW: *See* **World Wide Web**

WYSIWYG: What you see is what you get; simplified and largely inaccurate abbreviation for software that presents screen displays resembling a final product, such as a printed document or a Web page.

XYZ

XML: *See* **Extensible Markup Language**

APPENDIX A

HTML Quick Reference

IN THIS APPENDIX

- HTML Tags 862
- Character Entities 882

IPR2024-00143
Apple EX1011 Page 451

This appendix is a reference to the HTML tags you can use in your documents. Unless otherwise noted, all of the tags listed here are supported by both Microsoft Explorer 3.0 and Netscape Navigator 3.0. Note that some other browsers do not support all the tags listed.

The proposed HTML style sheet specification is also not covered here. Refer to the Netscape (`http://home.netscape.com/`) or Microsoft (`http://www.microsoft.com/`) Web sites for details on this and other late-breaking changes to the HTML standard.

HTML Tags

These tags are used to create a basic HTML page with text, headings, and lists. An (MS) beside the attribute indicates that it is only supported by Microsoft Internet Explorer.

Comments

`<!-- ... -->`	Creates a comment. Can also be used to hide JavaScript from browsers that do not support it.
`<COMMENT>...</COMMENT>`	The new official way of specifying comments.

Structure Tags

Tag	Attribute	Function
`<HTML>...</HTML>`		Encloses the entire HTML document.
`<HEAD>...</HEAD>`		Encloses the head of the HTML document.
`<BODY>...</BODY>`		Encloses the body (text and tags) of the HTML document.
	`BACKGROUND="..."`	The name or URL of the image to tile on the page background.
	`BGCOLOR="..."`	The color of the page background.
	`TEXT="..."`	The color of the page's text.
	`LINK="..."`	The color of unfollowed links.
	`ALINK="..."`	The color of activated links.
	`VLINK="..."`	The color of followed links.
	`BGPROPERTIES="..."`(MS)	Properties of background image. Currently allows only the value `FIXED`, which prevents the background image from scrolling.

Tag	Attribute	Function
	`TOPMARGIN="..."`(MS)	Top margin of the page, in pixels.
	`BOTTOMMARGIN="..."`(MS)	Bottom margin of the page, in pixels.
`<BASE>`		Indicates the full URL of the current document. This optional tag is used within `<HEAD>`.
	`HREF="..."`	The full URL of this document.
`<ISINDEX>`		Indicates that this document is a gateway script that allows searches.
	`PROMPT="..."`	The prompt for the search field.
	`ACTION="..."`	Gateway program to which the search string should be passed.
`<LINK>`		Indicates a link between this document and some other document. Generally used only by HTML-generating tools. `<LINK>` represents a link from this entire document to another, as opposed to `<A>`, which can create multiple links in the document. Not commonly used.
	`HREF="..."`	The URL of the document to call when the link is activated.
	`NAME="..."`	If the document is to be considered an anchor, the name of that anchor.
	`REL="..."`	The relationship between the linked-to document and the current document; for example, `"TOC"` or `"Glossary"`.
	`REV="..."`	A reverse relationship between the current document and the linked-to document.
	`URN="..."`	A Uniform Resource Number (URN), a unique identifier different from the URL in `HREF`.

continues

IPR2024-00143
Apple EX1011 Page 452

Tag	Attribute	Function
	TITLE="..."	The title of the linked-to document.
	METHODS="..."	The method by which the document is to be retrieved; for example, FTP, Gopher, and so on.
<META>		Indicates meta-information about this document (information about the document itself); for example, keywords for search engines, special HTTP headers to be used for retrieving this document, expiration dates, and so on. Meta-information is usually in the form of a key/value pair. Used in the document <HEAD>.
	HTTP-EQUIV="..."	Creates a new HTTP header field with the same name as the attribute's value; for example, HTTP-EQUIV="Expires". The value of that header is specified by the CONTENT attribute.
	NAME="..."	If meta-data is usually in the form of key/value pairs, NAME indicates the key; for example, Author or ID.
	CONTENT="..."	The content of the key/value pair (or of the HTTP header indicated by HTTP-EQUIV).
<NEXTID>		Indicates the "next" document to this one (as might be defined by a tool to manage HTML documents in series). <NEXTID> is considered obsolete.

Headings and Title

Tag	Attribute	Function
<H1>...</H1>		A first-level heading.
<H2>...</H2>		A second-level heading.
<H3>...</H3>		A third-level heading.

Tag	Attribute	Function
<H4>...</H4>		A fourth-level heading.
<H5>...</H5>		A fifth-level heading.
<H6>...</H6>		A sixth-level heading.
<TITLE>...</TITLE>		Indicates the title of the document. Used within <HEAD>.

All heading tags accept the following attribute:

	ALIGN="..."	Possible values are CENTER, LEFT, and RIGHT.

Paragraphs and Regions

Tag	Attribute	Function
<P>...</P>		A plain paragraph. The closing tag (</P>) is optional.
	ALIGN="..."	Align text to CENTER, LEFT, or RIGHT.
<DIV>...</DIV>		A region of text to be formatted.
	ALIGN="..."	Align text to CENTER, LEFT, or RIGHT.

Links

Tag	Attribute	Function
<A>...</A>		With the HREF attribute, creates a link to another document or anchor; with the NAME attribute, creates an anchor that can be linked to.
	HREF="..."	The URL of the document to be called when the link is activated.
	NAME="..."	The name of the anchor.
	REL="..."	The relationship between the linked-to document and the current document; for example, "TOC" or "Glossary". REL="..." is not commonly used.

continues

A HTML QUICK REFERENCE

IPR2024-00143
Apple EX1011 Page 453

Tag	Attribute	Function
	REV="..."	A reverse relationship between the current document and the linked-to document (not commonly used).
	URN="..."	A Uniform Resource Number (URN), a unique identifier different from the URL in HREF (not commonly used).
	TITLE="..."	The title of the linked-to document (not commonly used).
	METHODS="..."	The method by which the document is to be retrieved; for example, FTP, Gopher, and so on (not commonly used).
	TARGET="..."	The name of a frame that the linked document should appear in.

Lists

Tag	Attribute	Function
<OL>...</OL>		An ordered (numbered) list.
	TYPE="..."	The type of numerals to label the list. Possible values are A, a, I, i, and 1.
	START="..."	The value with which to start this list.
<UL>...</UL>		An unordered (bulleted) list.
	TYPE="..."	The bullet dingbat to use to mark list items. Possible values are DISC, CIRCLE (or ROUND), and SQUARE.
<MENU>...</MENU>		A menu list of items.
<DIR>...</DIR>		A directory listing; items are generally smaller than 20 characters.
<LI>		A list item for use with <OL>, <UL>, <MENU>, or <DIR>.

Tag	Attribute	Function
	TYPE="..."	The type of bullet or number to label this item with. Possible values are DISC, CIRCLE (or ROUND) SQUARE, A, a, I, i, and 1.
	VALUE="..."	The numeric value this list item should have (affects this item and all below it in <OL> lists).
<DL>...</DL>		A definition or glossary list.
	COMPACT	The COMPACT attribute specifies formatting that takes less whitespace to present.
<DT>		A definition term as part of definition list.
<DD>		The corresponding definition to a definition term as part of definition list.

Character Formatting

Tag	Attribute	Function
<EM>...</EM>		Emphasis (usually italic).
<STRONG>...</STRONG>		Stronger emphasis (usually bold).
<CODE>...</CODE>		Code sample (usually Courier).
<KBD>...</KBD>		Text to be typed (usually Courier).
<VAR>...</VAR>		A variable or placeholder for some other value.
<SAMP>...</SAMP>		Sample text (not commonly used).
<DFN>...</DFN>		A definition of a term.
<CITE>...</CITE>		A citation.
<B>...</B>		Boldface text.
<I>...</I>		Italic text.
<TT>...</TT>		Typewriter (monospaced) font.

continues

IPR2024-00143
Apple EX1011 Page 454

Tag	Attribute	Function
<PRE>...</PRE>		Preformatted text (exact line endings and spacing will be preserved—usually rendered in a monospaced font).
<BIG>...</BIG>		Text is slightly larger than normal.
<SMALL>...</SMALL>		Text is slightly smaller than normal.
_{...}		Subscript.
^{...}		Superscript.
<STRIKE>...</STRIKE>		Puts a strikethrough line in text.

Other Elements

Tag	Attribute	Function
<HR>		A horizontal rule line.
	SIZE="..."	The thickness of the rule, in pixels.
	WIDTH="..."	The width of the rule, in pixels or as a percentage of the document width.
	ALIGN="..."	How the rule line will be aligned on the page. Possible values are LEFT, RIGHT, and CENTER.
	NOSHADE	Causes the rule line to be drawn as a solid line instead of a transparent bevel.
	COLOR="..." (MS)	Color of the horizontal rule.
 		A line break.
	CLEAR="..."	Causes the text to stop flowing around any images. Possible values are RIGHT, LEFT, and ALL.
<NOBR>...</NOBR>		Causes the enclosed text not to wrap at the edge of the page.
<WBR>		Wraps the text at this point only if necessary.

Tag	Attribute	Function
<BLOCKQUOTE>... </BLOCKQUOTE>		Used for long quotes or citations.
<ADDRESS>... </ADDRESS>		Used for signatures or general information about a document's author.
<CENTER>... </CENTER>		Centers text or images.
<BLINK>...</BLINK>		Causes the enclosed text to blink in an irritating manner.
<FONT>...</FONT>		Changes the size of the font for the enclosed text.
	SIZE="..."	The size of the font, from 1 to 7. Default is 3. Can also be specified as a value relative to the current size; for example, +2.
	COLOR="..."	Changes the color of the text.
	FACE="..."	Name of font to use if it can be found on the user's system. Multiple font names can be separated by commas, and the first font on the list that can be found will be used.
<BASEFONT>		Sets the default size of the font for the current page.
	SIZE="..."	The default size of the font, from 1 to 7. Default is 3.

Images, Sounds, and Embedded Media

Tag	Attribute	Function
<IMG>		Inserts an inline image into the document.
	ISMAP	This image is a clickable image map.
	SRC="..."	The URL of the image.

continues

A HTML QUICK REFERENCE

IPR2024-00143
Apple EX1011 Page 455

Tag	Attribute	Function
	ALT="..."	A text string that will be displayed in browsers that cannot support images.
	ALIGN="..."	Determines the alignment of the given image. If LEFT or RIGHT, the image is aligned to the left or right column, and all following text flows beside that image. All other values such as TOP, MIDDLE, and BOTTOM, or the Netscape-only TEXTTOP, ABSMIDDLE, BASELINE, and ABSBOTTOM determine the vertical alignment of this image with other items in the same line.
	VSPACE="..."	The space between the image and the text above or below it.
	HSPACE="..."	The space between the image and the text to its left or right.
	WIDTH="..."	The width, in pixels, of the image. If WIDTH is not the actual width, the image is scaled to fit.
	HEIGHT="..."	The height, in pixels, of the image. If HEIGHT is not the actual height, the image is scaled to fit.
	BORDER="..."	Draws a border of the specified value in pixels to be drawn around the image. In the case of images that are also links, BORDER changes the size of the default link border.
	LOWSRC="..."	The path or URL of an image that will be loaded first, before the image specified in SRC. The value of LOWSRC is usually a smaller or lower resolution version of the actual image.
	USEMAP="..."	The name of an image map specification for client-side image mapping. Used with <MAP> and <AREA>.

Tag	Attribute	Function
	DYNSRC="..." (MS)	The address of a video clip or VRML world (dynamic source).
	CONTROLS (MS)	Used with DYNSRC to display a set of playback controls for inline video.
	LOOP="..." (MS)	The number of times a video clip will loop. (-1, or INFINITE, means to loop indefinitely.)
	START="..." (MS)	When a DYNSRC video clip should start playing. Valid options are FILEOPEN (play when page is displayed) or MOUSEOVER (play when mouse cursor passes over the video clip).
<BGSOUND> (MS)		Plays a sound file as soon as the page is displayed.
	SRC="..."	The URL of the WAV, AU, or MIDI sound file to embed.
	LOOP="..." (MS)	The number of times a video clip will loop. (-1, or INFINITE, means to loop indefinitely.)
<SCRIPT>		An interpreted script program.
	LANGUAGE="..."	Currently only JAVASCRIPT is supported by Netscape. Both JAVASCRIPT and VBSCRIPT are supported by Microsoft.
	SRC="..."	Specifies the URL of a file that includes the script program.
<OBJECT>		Inserts an image, video, Java applet, or ActiveX control into a document.

NOTE

Usage of the <OBJECT> tag is not yet finalized. Check http://www.w3.org/ for the latest attributes supported by the HTML 3.2 standard.

A HTML QUICK REFERENCE

IPR2024-00143
Apple EX1011 Page 456

Tag	Attribute	Function
<APPLET>		Inserts a self-running Java applet.
	CLASS="..."	The name of the applet.
	SRC="..."	The URL of the directory where the compiled applet can be found (should end in a slash as in http://mysite/myapplets/). Do not include the actual applet name, which is specified with the CLASS attribute.
	ALIGN="..."	Indicates how the applet should be aligned with any text that follows it. Current values are TOP, MIDDLE, and BOTTOM.
	WIDTH="..."	The width of the applet output area, in pixels.
	HEIGHT="..."	The height of the applet output area, in pixels.
<PARAM>		Program-specific parameters. (Always occurs within <APPLET> or <OBJECT> tags.)
	NAME="..."	The type of information being given to the applet or ActiveX control.
	VALUE="..."	The actual information to be given to the applet or ActiveX control.
	REF="..."	Indicates that this <PARAM> tag includes the address or location of the object.
<EMBED> (Netscape only!)		Embeds a file to be read or displayed by a plug-in application.

NOTE

In addition to the following standard attributes, you can specify applet-specific attributes to be interpreted by the plug-in that displays the embedded object.

Tag	Attribute	Function
	SRC="..."	The URL of the file to embed.
	WIDTH="..."	The width of the embedded object in pixels.
	HEIGHT="..."	The height of the embedded object in pixels.
	ALIGN="..."	Determines the alignment of the media window. Values are the same as for the <IMG> tag.
	VSPACE="..."	The space between the media and the text above or below it.
	HSPACE="..."	The space between the media and the text to its left or right.
	BORDER="..."	Draws a border of the specified size in pixels to be drawn around the media.
<NOEMBED>...</NOEMBED>		Alternate text or images to be shown to users who do not have a plug-in installed.
<MAP>...</MAP>		A client-side image map, referenced by <IMG USEMAP="...">. Includes one or more <AREA> tags.
<AREA>		Defines a clickable link within a client-side image map.
	SHAPE="..."	The shape of the clickable area. Currently, only RECT is supported.
	COORDS="..."	The left, top, right, and bottom coordinates of the clickable region within an image.
	HREF="..."	The URL that should be loaded when the area is clicked.
	NOHREF	Indicates that no action should be taken when this area of the image is clicked.

IPR2024-00143
Apple EX1011 Page 457

Forms

Tag	Attribute	Function
<FORM>...</FORM>		Indicates an input form.
	ACTION="..."	The URL of the script to process this form input.
	METHOD="..."	How the form input will be sent to the gateway on the server side. Possible values are GET and POST.
	ENCTYPE="..."	Normally has the value application/x-www-form-urlencoded. For file uploads, use multipart/form-data.
	NAME="..."	A name by which JavaScript scripts can refer to the form.
<INPUT>		An input element for a form.
	TYPE="..."	The type for this input widget. Possible values are CHECKBOX, HIDDEN, RADIO, RESET, SUBMIT, TEXT, SEND FILE, or IMAGE.
	NAME="..."	The name of this item, as passed to the gateway script as part of a name/value pair.
	VALUE="..."	For a text or hidden widget, the default value; for a checkbox or radio button, the value to be submitted with the form; for Reset or Submit buttons, the label for the button itself.
	SRC="..."	The source file for an image.
	CHECKED	For checkboxes and radio buttons, indicates that the widget is checked.
	SIZE="..."	The size, in characters, of a text widget.
	MAXLENGTH="..."	The maximum number of characters that can be entered into a text widget.

Tag	Attribute	Function
	ALIGN="..."	For images in forms, determines how the text and image will align (same as with the <IMG> tag).
<TEXTAREA>...</TEXTAREA>		Indicates a multiline text entry form element. Default text can be included.
	NAME="..."	The name to be passed to the gateway script as part of the name/value pair.
	ROWS="..."	The number of rows this text area displays.
	COLS="..."	The number of columns (characters) this text area displays.
	WRAP="..."	Controls text wrapping. Possible values are OFF, VIRTUAL, and PHYSICAL.
<SELECT>...</SELECT>		Creates a menu or scrolling list of possible items.
	NAME="..."	The name that is passed to the gateway script as part of the name/value pair.
	SIZE="..."	The number of elements to display. If SIZE is indicated, the selection becomes a scrolling list. If no SIZE is given, the selection is a pop-up menu.
	MULTIPLE	Allows multiple selections from the list.
<OPTION>		Indicates a possible item within a <SELECT> element.
	SELECTED	With this attribute included, the <OPTION> will be selected by default in the list.
	VALUE="..."	The value to submit if this <OPTION> is selected when the form is submitted.

IPR2024-00143
Apple EX1011 Page 458

Tables

Tag	Attribute	Function
<TABLE>...</TABLE>		Creates a table that can contain a caption (<CAPTION>) and any number of rows (<TR>).
	BORDER="..."	Indicates whether the table should be drawn with or without a border. In Netscape, BORDER can also have a value indicating the width of the border.
	CELLSPACING="..."	The amount of space between the cells in the table.
	CELLPADDING="..."	The amount of space between the edges of the cell and its contents.
	WIDTH="..."	The width of the table on the page, in either exact pixel values or as a percentage of page width.
	ALIGN="..." (MS)	Alignment (works like IMG ALIGN). Values are LEFT or RIGHT.
	BGCOLOR="..."	Background color of all cells in the table that do not contain their own BACKGROUND or BGCOLOR attribute.
	BACKGROUND="..." (MS)	Background image to tile within all cells in the table that do not contain their own BACKGROUND or BGCOLOR attribute.
	BORDERCOLOR="..." (MS)	Border color (used with BORDER="...").

Tag	Attribute	Function
	BORDERCOLORLIGHT="..." (MS)	Color for light part of 3D-look borders (used with BORDER="...").
	BORDERCOLORDARK="..." (MS)	Color for dark part of 3D-look borders (used with BORDER="...").
	VALIGN="..." (MS)	Alignment of text within the table. Values are TOP and BOTTOM.
	FRAME="..." (MS)	Controls which external borders will appear around a table. Values are void (no frames), above (top border only), below (bottom border only), hsides (top and bottom), lhs (left side), rhs (right side), vsides (left and right sides), and box (all sides).
	RULES="..." (MS)	Controls which internal borders appear in the table. Values are none, basic (rules between THEAD, TBODY, and TFOOT only), rows (horizontal borders only), cols (vertical borders only), and all.
<CAPTION>...</CAPTION>		The caption for the table.
	ALIGN="..."	The position of the caption. Possible values are TOP and BOTTOM.

continues

A HTML QUICK REFERENCE

IPR2024-00143
Apple EX1011 Page 459

Tag	Attribute	Function
<TR>...</TR>		Defines a table row, containing headings and data (<TR> and <TH> tags).
	ALIGN="..."	The horizontal alignment of the contents of the cells within this row. Possible values are LEFT, RIGHT, and CENTER.
	VALIGN="..."	The vertical alignment of the contents of the cells within this row. Possible values are TOP, MIDDLE, BOTTOM, and BASELINE.
	BGCOLOR="..."	Background color of all cells in the row that do not contain their own BACKGROUND or BGCOLOR attributes.
	BACKGROUND="..."(MS)	Background image to tile within all cells in the row that do not contain their own BACKGROUND or BGCOLOR attributes.
	BORDERCOLOR="..."(MS)	Border color (used with BORDER="...").
	BORDERCOLORLIGHT="..."(MS)	Color for light part of 3D-look borders (used with BORDER="...").
	BORDERCOLORDARK="..."(MS)	Color for dark part of 3D-look borders (used with BORDER="...").
<TH>...</TH>		Defines a table heading cell.

Tag	Attribute	Function
	ALIGN="..."	The horizontal alignment of the contents of the cell. Possible values are LEFT, RIGHT, and CENTER.
	VALIGN="..."	The vertical alignment of the contents of the cell. Possible values are TOP, MIDDLE, BOTTOM, and BASELINE.
	ROWSPAN="..."	The number of rows this cell will span.
	COLSPAN="..."	The number of columns this cell will span.
	NOWRAP	Does not automatically wrap the contents of this cell.
	WIDTH="..."	The width of this column of cells, in exact pixel values or as a percentage of the table width.
	BGCOLOR="..."	Background color of the cell.
	BACKGROUND="..." (MS)	Background image to tile within the cell.
	BORDERCOLOR="..." (MS)	Border color (used with BORDER="...").
	BORDERCOLORLIGHT="..." (MS)	Color for light part of 3D-look borders (used with BORDER="...").
	BORDERCOLORDARK="..." (MS)	Color for dark part of 3D-look borders (used with BORDER="...").
<TD>...</TD>		Defines a table data cell.

continues

A HTML QUICK REFERENCE

IPR2024-00143
Apple EX1011 Page 460

Tag	Attribute	Function
	ALIGN="..."	The horizontal alignment of the contents of the cell. Possible values are LEFT, RIGHT, and CENTER.
	VALIGN="..."	The vertical alignment of the contents of the cell. Possible values are TOP, MIDDLE, BOTTOM, and BASELINE.
	ROWSPAN="..."	The number of rows this cell will span.
	COLSPAN="..."	The number of columns this cell will span.
	NOWRAP	Does not automatically wrap the contents of this cell.
	WIDTH="..."	The width of this column of cells, in exact pixel values or as a percentage of the table width.
	BGCOLOR="..."	Background color of the cell.
	BACKGROUND="..." (MS)	Background image to tile within the cell.
	BORDERCOLOR="..." (MS)	Border color (used with BORDER="...").
	BORDERCOLORLIGHT="..." (MS)	Color for light part of 3D-look borders (used with BORDER="...").
	BORDERCOLORDARK="..." (MS)	Color for dark part of 3D-look borders (used with BORDER="...").

Frames

Tag	Attribute	Function
<FRAMESET>... </FRAMESET>		Divides the main window into a set of frames that can each display a separate document.
	ROWS="..."	Splits the window or frameset vertically into a number of rows specified by a number (such as 7), a percentage of the total window width (such as 25%), or an asterisk (*) indicating that a frame should take up all the remaining space or divide the space evenly between frames (if multiple * frames are specified).
	COLS="..."	Works similar to ROWS, except that the window or frameset is split horizontally into columns.
	BORDER="..."	Size of frame border in pixels (0 turns off borders). This tag is Netscape-specific; Microsoft IE uses FRAMEBORDER and FRAMESPACING instead.
	FRAMEBORDER="..." (MS)	Specifies whether to display a border for a frame. Options are YES and NO.
	FRAMESPACING="..." (MS)	Space between frames, in pixels.
<FRAME>		Defines a single frame within a <FRAMESET>.
	SRC="..."	The URL of the document to be displayed in this frame.

continues

IPR2024-00143
Apple EX1011 Page 461

Tag	Attribute	Function
	NAME="..."	A name to be used for targeting this frame with the TARGET attribute in <A HREF> links.
	MARGINWIDTH="..."	The amount of space to leave to the left and right side of a document within a frame, in pixels.
	MARGINHEIGHT="..."	The amount of space to leave above and below a document within a frame, in pixels.
	SCROLLING="..."	Determines whether a frame has scrollbars. Possible values are YES, NO, and AUTO.
	NORESIZE	Prevents the user from resizing this frame (and possibly adjacent frames) with the mouse.
<NOFRAME>...</NOFRAME>		Provides an alternative document body in <FRAMESET> documents for browsers that do not support frames (usually encloses <BODY>...</BODY>).

Character Entities

Table A.1 contains the possible numeric and character entities for the ISO-Latin-1 (ISO8859-1) character set. Where possible, the character is shown.

NOTE

Not all browsers can display all characters, and some browsers might even display characters different from those that appear in the table. Newer browsers seem to have a better track record for handling character entities, but be sure to test your HTML files extensively with multiple browsers if you intend to use these entities.

Table A.1. ISO-Latin-1 character set.

Character	*Numeric Entity*	*Character Entity (if any)*	*Description*
	&#00;–&#08;		Unused
			Horizontal tab
	Line feed
	&#11;–&#31;		Unused
			Space
!	!		Exclamation mark
"	"	"	Quotation mark
#	#		Number sign
$	$		Dollar sign
%	%		Percent sign
&	&	&	Ampersand
'	'		Apostrophe
(	(		Left parenthesis
)	)		Right parenthesis
*	*		Asterisk
+	+		Plus sign
,	,		Comma
-	-		Hyphen
.	.		Period (fullstop)
/	/		Solidus (slash)
0–9	0–9		Digits 0–9
:	:		Colon
;	;		Semicolon
<	<	<	Less than
=	=		Equal sign
>	>	>	Greater than
?	?		Question mark
@	@		Commercial "at"
A–Z	A–Z		Letters A–Z

continues

IPR2024-00143
Apple EX1011 Page 462

Table A.1. continued

Character	*Numeric Entity*	*Character Entity (if any)*	*Description*
[	[		Left square bracket
\	\		Reverse solidus (backslash)
]	]		Right square bracket
^	^		Caret
—	_		Horizontal bar
`	`		Grave accent
a–z	a–z		Letters a–z
{	{		Left curly brace
|	&#124		Vertical bar
}	}		Right curly brace
~	~		Tilde
	&#127;–		Unused
¡	¡	¡	Inverted exclamation
¢	¢	¢	Cent sign
£	£	£	Pound sterling
¤	¤	¤	General currency sign
¥	¥	¥	Yen sign
¦	¦	¦ or &brkbar;	Broken vertical bar
§	§	§	Section sign
¨	¨	¨	Umlaut (diaeresis)
©	©	© (Netscape only)	Copyright
ª	ª	ª	Feminine ordinal
«	«	«	Left angle quote, guillemot left
¬	¬	¬	Not sign

Character	*Numeric Entity*	*Character Entity (if any)*	*Description*
-			Soft hyphen
®	®	® (Netscape only)	Registered trademark
¯	¯	¯	Macron accent
°	°	°	Degree sign
±	±	±	Plus or minus
²	²	²	Superscript two
³	³	³	Superscript three
´	´	´	Acute accent
µ	µ	µ	Micro sign
¶	¶	¶	Paragraph sign
·	·	·	Middle dot
¸	¸	¸	Cedilla
¹	¹	¹	Superscript one
º	º	º	Masculine ordinal
»	»	»	Right angle quote, guillemot right
¼	¼	¼	Fraction one-fourth
½	½	½	Fraction one-half
¾	¾	¾	Fraction three-fourths
¿	¿	&iquest	Inverted question mark
À	À	À	Capital A, grave accent
Á	Á	Á	Capital A, acute accent
Â	Â	Â	Capital A, circumflex accent

continues

A HTML QUICK REFERENCE

IPR2024-00143
Apple EX1011 Page 463

Table A.1. continued

Character	*Numeric Entity*	*Character Entity (if any)*	*Description*
Ã	Ã	Ã	Capital A, tilde
Ä	Ä	Ä	Capital A, dieresis or umlaut mark
Å	Å	Å	Capital A, ring
Æ	Æ	Æ	Capital AE diphthong (ligature)
Ç	Ç	Ç	Capital C, cedilla
È	È	È	Capital E, grave accent
É	É	É	Capital E, acute accent
Ê	Ê	Ê	Capital E, circumflex accent
Ë	Ë	Ë	Capital E, dieresis or umlaut mark
Ì	Ì	Ì	Capital I, grave accent
Í	Í	Í	Capital I, acute accent
Î	Î	Î	Capital I, circumflex accent
Ï	Ï	Ï	Capital I, dieresis or umlaut mark
Ð	Ð	Ð	Capital Eth, Icelandic
Ñ	Ñ	Ñ	Capital N, tilde
Ò	Ò	Ò	Capital O, grave accent

Character	*Numeric Entity*	*Character Entity (if any)*	*Description*
Ó	Ó	Ó	Capital O, acute accent
Ô	Ô	Ô	Capital O, circumflex accent
Õ	Õ	Õ	Capital O, tilde
Ö	Ö	Ö	Capital O, dieresis or umlaut mark
×	×	×	Multiply sign
Ø	Ø	Ø	Capital O, slash
Ù	Ù	Ù	Capital U, grave accent
Ú	Ú	Ú	Capital U, acute accent
Û	Û	Û	Capital U, circumflex accent
Ü	Ü	Ü	Capital U, dieresis or umlaut mark
Ý	Ý	Ý	Capital Y, acute accent
Þ	Þ	Þ	Capital THORN, Icelandic
ß	ß	ß	Small sharp s, German (sz ligature)
à	à	à	Small a, grave accent
á	á	á	Small a, acute accent
â	â	â	Small a, circumflex accent

continues

IPR2024-00143
Apple EX1011 Page 464

Table A.1. continued

Character	*Numeric Entity*	*Character Entity (if any)*	*Description*
ã	ã	ã	Small a, tilde
ä	ä	ä	Small a, dieresis or umlaut mark
å	å	å	Small a, ring
æ	æ	æ	Small ae diphthong (ligature)
ç	ç	ç	Small c, cedilla
è	è	è	Small e, grave accent
é	é	é	Small e, acute accent
ê	ê	ê	Small e, circumflex accent
ë	ë	ë	Small e, dieresis or umlaut mark
ì	ì	ì	Small i, grave accent
í	í	í	Small i, acute accent
î	î	î	Small i, circumflex accent
ï	ï	ï	Small i, dieresis or umlaut mark
ð	ð	ð	Small eth, Icelandic
ñ	ñ	ñ	Small n, tilde
ò	ò	ò	Small o, grave accent
ó	ó	ó	Small o, acute accent
ô	ô	ô	Small o, circumflex accent

Character	*Numeric Entity*	*Character Entity (if any)*	*Description*
õ	õ	õ	Small o, tilde
ö	ö	ö	Small o, dieresis or umlaut mark
÷	÷	÷	Division sign
ø	ø	ø	Small o, slash
ù	ù	ù	Small u, grave accent
ú	ú	ú	Small u, acute accent
û	û	û	Small u, circumflex accent
ü	ü	ü	Small u, dieresis or umlaut mark
ý	ý	ý	Small y, acute accent
þ	þ	þ	Small thorn, Icelandic
ÿ	ÿ	ÿ	Small y, dieresis or umlaut mark

IPR2024-00143
Apple EX1011 Page 465

APPENDIX B

HTML 3.2 Reference Specification

by Dave Raggett (dsr@w3.org)

IN THIS APPENDIX

- Abstract 892
- Contents 892
- Introduction to HTML 3.2 892
- The HEAD Element 895
- The BODY Element 899
- Text-Level Elements 913
- Sample SGML Open Catalog for HTML 3.2 927
- SGML Declaration for HTML 3.2 927
- HTML 3.2 Document Type Definition 929
- Character Entities for ISO Latin-1 940
- Table of Printable Latin-1 Character Codes 942
- Acknowledgments 943
- Further Reading 943

IPR2024-00143
Apple EX1011 Page 466

Status of This Document

This document has been reviewed by W3C members and other interested parties and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

A list of current W3C Recommendations and other technical documents can be found at `http://www.w3.org/pub/WWW/TR/">http://www.w3.org/pub/WWW/TR/`.

Abstract

The Hypertext Markup Language (HTML) is a simple markup language used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. This specification defines HTML Version 3.2. HTML 3.2 aims to capture recommended practice as of early '96 and as such to be used as a replacement for HTML 2.0 (RFC 1866).

Contents

- Introduction to HTML 3.2
- HTML as an SGML Application
- The Structure of HTML Documents
- The HEAD Element and Its Children
- The BODY Element and Its Children
- Sample SGML Open Catalog for HTML 3.2
- SGML Declaration for HTML 3.2
- HTML 3.2 Document Type Definition
- Character Entities for ISO Latin-1
- Table of Printable Latin-1 Character Codes
- Acknowledgments
- Further Reading...

Introduction to HTML 3.2

HTML 3.2 is W3C's specification for HTML, developed in early '96 together with vendors including IBM, Microsoft, Netscape Communications Corporation, Novell, SoftQuad, Spyglass, and Sun Microsystems. HTML 3.2 adds widely deployed features such as tables, applets

and text flow around images, while providing full backwards compatibility with the existing standard HTML 2.0.

W3C is continuing to work with vendors on extensions for accessibility features, multimedia objects, scripting, style sheets, layout, forms, math, and internationalization. W3C plans on incorporating this work in further versions of HTML.

HTML as an SGML Application

HTML 3.2 is an SGML application conforming to International Standard ISO 8879—Standard Generalized Markup Language. As an SGML application, the syntax of conforming HTML 3.2 documents is defined by the combination of the SGML declaration and the document type definition (DTD). This specification defines the intended interpretation of HTML 3.2 elements and places further constraints on the permitted syntax that are otherwise inexpressible in the DTD.

The SGML rules for record boundaries are tricky. In particular, a record end immediately following a start tag should be discarded. For example:

```
<P>
Text
```

is equivalent to

```
<P>Text
```

Similarly, a record end immediately preceding an end tag should be discarded. For example:

```
Text
</P>
```

is equivalent to

```
Text</P>
```

Except within literal text (for example, the PRE element), HTML treats contiguous sequences of white space characters as being equivalent to a single space character (ASCII decimal 32). These rules allow authors considerable flexibility when editing the marked-up text directly. Note that future revisions to HTML may allow for the interpretation of the horizontal tab character (ASCII decimal 9) with respect to a tab rule defined by an associated style sheet.

SGML entities in PCDATA content or in CDATA attributes are expanded by the parser; for example, `é` is expanded to the ISO Latin-1 character decimal 233 (a lowercase letter *e* with an acute accent). This could also have been written as a named character entity, for example, `é`. The `&` character can be included in its own right using the named character entity `&`.

HTML allows CDATA attributes to be unquoted provided the attribute value contains only letters (*a* to *z* and *A* to *Z*), digits (0 to 9), hyphens (ASCII decimal 45), or periods (ASCII decimal 46). Attribute values can be quoted using double or single quote marks (ASCII decimal 34 and 39, respectively). Single quote marks can be included within the attribute value when the value is delimited by double quote marks, and vice versa.

IPR2024-00143
Apple EX1011 Page 467

Note that some user agents require attribute minimization for the following attributes: COMPACT, ISMAP, CHECKED, NOWRAP, NOSHADE, and NOHREF. These user agents don't accept syntax such as COMPACT=COMPACT or ISMAP=ISMAP although this is legitimate according to the HTML 3.2 DTD.

The SGML declaration and the DTD for use with HTML 3.2 are given in appendices. Further guidelines for parsing HTML are given in WD-html-lex.

The Structure of HTML Documents

HTML 3.2 documents start with a <!DOCTYPE> declaration followed by an HTML element containing a HEAD and then a BODY element:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN>
<HTML>
<HEAD>
<TITLE>A study of population dynamics</TITLE>
... other head elements
</HEAD>
<BODY>
... document body
</BODY>
</HTML>
```

In practice, the HTML HEAD and BODY start and end tags can be omitted from the markup as these can be inferred in all cases by parsers conforming to the HTML 3.2 DTD.

Every conforming HTML 3.2 document **must** start with the <!DOCTYPE> declaration that is needed to distinguish HTML 3.2 documents from other versions of HTML. The HTML specification is not concerned with storage entities. As a result, it is not required that the document type declaration reside in the same storage entity (that is, file). A Web site may choose to dynamically prepend HTML files with the document type declaration if it is known that all such HTML files conform to the HTML 3.2 specification.

Every HTML 3.2 document must also include the descriptive title element. A minimal HTML 3.2 document thus looks like this:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN>
<TITLE>A study of population dynamics</TITLE>
```

NOTE

The word "Final" replaces "Draft" now that the HTML 3.2 specification has been ratified by the W3C member organizations.

The HEAD Element

This contains the document head, but you can always omit both the start and end tags for HEAD. The contents of the document head are an unordered collection of the following elements:

- The TITLE element
- The STYLE element
- The SCRIPT element
- The ISINDEX element
- The BASE element
- The META element
- The LINK element

```
<!ENTITY % head.content "TITLE & ISINDEX? & BASE?">
<!ENTITY % head.misc "SCRIPT|STYLE|META|LINK">

<!ELEMENT HEAD O O  (%head.content) +(%head.misc)>
```

The %head.misc entity is used to allow the associated elements to occur multiple times at arbitrary positions within the HEAD. The following elements can be part of the document head:

- TITLE defines the document title, and is always needed.
- ISINDEX is for simple keyword searches; see PROMPT attribute.
- BASE defines base URL for resolving relative URLs.
- SCRIPT is reserved for future use with scripting languages.
- STYLE is reserved for future use with style sheets.
- META is used to supply meta info as name/value pairs.
- LINK is used to define relationships with other documents.

TITLE, SCRIPT and STYLE are containers and require both start and end tags. The other elements are not containers so that end tags are forbidden. Note that conforming browsers won't render the contents of SCRIPT and STYLE elements.

TITLE

```
<!ELEMENT TITLE - -  (#PCDATA)* -(%head.misc)>
```

Every HTML 3.2 document **must** have exactly one TITLE element in the document's HEAD. It provides an advisory title that can be displayed in a user agent's window caption and so on. The content model is PCDATA. As a result, character entities can be used for accented characters and to escape special characters such as & and <. Markup is not permitted in the content of a TITLE element.

IPR2024-00143
Apple EX1011 Page 468

Example TITLE element:

```
<TITLE>A study of population dynamics</TITLE>
```

STYLE and SCRIPT

```
<!ELEMENT STYLE  - - CDATA -- placeholder for style info -->
<!ELEMENT SCRIPT - - CDATA -- placeholder for script statements -->
```

These are placeholders for the introduction of style sheets and client-side scripts in future versions of HTML. User agents should hide the contents of these elements.

These elements are defined with CDATA as the content type. As a result they may contain only SGML characters. All markup characters or delimiters are ignored and passed as data to the application, except for ETAGO (</) delimiters followed immediately by a name character [a-zA-Z]. This means that the element's end tag (or that of an element in which it is nested) is recognized, while an error occurs if the ETAGO is invalid.

ISINDEX

```
<!ELEMENT ISINDEX - O EMPTY>
<!ATTLIST ISINDEX
        prompt CDATA #IMPLIED -- prompt message -->
```

The ISINDEX element indicates that the user agent should provide a single-line text input field for entering a query string. There are no restrictions on the number of characters that can be entered. The PROMPT attribute can be used to specify a prompt string for the input field, for example:

```
<ISINDEX PROMPT="Search Phrase">
```

The semantics for ISINDEX are currently well defined only when the base URL for the enclosing document is an HTTP URL. Typically, when the user presses the Enter (return) key, the query string is sent to the server identified by the base URL for this document. For example, if the query string entered is "ten green apples" and the base URL is

```
http://www.acme.com/
```

then the query generated is

```
http://www.acme.com/?ten+green+apples"
```

Note that space characters are mapped to + characters and that normal URL character escaping mechanisms apply. For further details, see the HTTP specification.

NOTE

Note that in practice, the query string is restricted to Latin-1 as there is no current mechanism for the URL to specify a character set for the query.

BASE

```
<!ELEMENT BASE - O EMPTY>
<!ATTLIST BASE
        href %URL  #REQUIRED
        >
```

The BASE element gives the base URL for dereferencing relative URLs, using the rules given by the URL specification, for example,

```
<BASE href="http://www.acme.com/intro.html">
    ...
    <IMG SRC="icons/logo.gif">
```

The image is dereferenced to

```
http://www.acme.com/icons/logo.gif
```

In the absence of a BASE element, the document URL should be used. Note that this is not necessarily the same as the URL used to request the document, as the base URL may be overridden by an HTTP header accompanying the document.

META

```
<!ELEMENT META - O EMPTY    -- Generic Metainformation -->
<!ATTLIST META
        http-equiv  NAME    #IMPLIED  -- HTTP response header name  --
        name        NAME    #IMPLIED  -- metainformation name       --
        content     CDATA   #REQUIRED -- associated information     --
        >
```

The META element can be used to include name/value pairs describing properties of the document, such as author, expire date, a list of key words, and so on. The NAME attribute specifies the property name, while the CONTENT attribute specifies the property value, for example,

```
<META NAME="Author" CONTENT="Dave Raggett">
```

The HTTP-EQUIV attribute can be used in place of the NAME attribute and has a special significance when documents are retrieved via the Hypertext Transfer Protocol (HTTP). HTTP servers may use the property name specified by the HTTP-EQUIV attribute to create an RFC 822 style header in the HTTP response. This can't be used to set certain HTTP headers, though; see the HTTP specification for details.

```
<META HTTP-EQUIV="Expires" CONTENT="Tue, 20 Aug 1996 14:25:27 GMT">
```

will result in the HTTP header:

```
Expires: Tue, 20 Aug 1996 14:25:27 GMT
```

This can be used by caches to determine when to fetch a fresh copy of the associated document.

IPR2024-00143
Apple EX1011 Page 469

LINK

LINK provides a media-independent method for defining relationships with other documents and resources. LINK has been part of HTML since the very early days, although few browsers as yet take advantage of it (most still ignore LINK elements).

LINK elements can be used *in principle* in the following ways:

- For document-specific navigation toolbars or menus
- To control how collections of HTML files are rendered into printed documents
- For linking associated resources such as style sheets and scripts
- To provide alternative forms of the current document

```
<!ELEMENT LINK - O EMPTY>
<!ATTLIST LINK
        href    %URL    #IMPLIED    -- URL for linked resource --
        rel     CDATA   #IMPLIED    -- forward link types --
        rev     CDATA   #IMPLIED    -- reverse link types --
        title   CDATA   #IMPLIED    -- advisory title string --
        >
```

`href` Specifies a URL designating the linked resource.

`rel` The forward relationship also known as the "link type." It specifies a named relationship from the enclosing document to the resource specified by the HREF attribute. HTML link relationships are as yet unstandardized, although some conventions have been established.

`rev` This defines a reverse relationship. A link from document A to document B with REV=*relation* expresses the same relationship as a link from B to A with REL=*relation*. REV=made is sometimes used to identify the document author, either the author's e-mail address with a mailto URL, or a link to the author's home page.

`title` An advisory title for the linked resource.

Here are some proposed relationship values:

`rel=top`

The link references the top of a hierarchy, for example, the first or cover page in a collection.

`rel=contents`

The link references a document serving as a table of contents.

`rel=index`

The link references a document providing an index for the current document.

`rel=glossary`

The link references a document providing a glossary of terms that are relevant to the current document.

`rel=copyright`

The link references a copyright statement for the current document.

`rel=next`

The link references the next document to visit in a guided tour. It can be used, for example, to preload the next page.

`rel=previous`

The link references the previous document in a guided tour.

`rel=help`

The link references a document offering help, for example, describing the wider context and offering further links to relevant documents. This is aimed at reorienting users who have lost their way.

`rel=search`

The link references a page for searching material related to a collection of pages

Example LINK elements:

```
<LINK REL=Contents HREF=toc.html>
<LINK REL=Previous HREF=doc31.html>
<LINK REL=Next HREF=doc33.html>
<LINK REL=Chapter REV=Contents HREF=chapter2.html>
```

The BODY Element

This contains the document body. Both start and end tags for BODY may be omitted. The body can contain a wide range of elements:

- Headings (H1–H6)
- The ADDRESS element
- Block-level elements
- Text-level elements

The key attributes are BACKGROUND, BGCOLOR, TEXT, LINK, VLINK, and ALINK. These can be used to set a repeating background image, plus background and foreground colors for normal text and hypertext links.

```
<!ENTITY % body.content "(%heading | %text | %block | ADDRESS)*">
<!ENTITY % color "CDATA" -- a color specification: #HHHHHH @@ details? -->
```


IPR2024-00143
Apple EX1011 Page 470

```
<!ENTITY % body-color-attrs "
        bgcolor %color #IMPLIED
        text %color #IMPLIED
        link %color #IMPLIED
        vlink %color #IMPLIED
        alink %color #IMPLIED
        ">
<!ELEMENT BODY O O  %body.content>
<!ATTLIST BODY
        background %URL #IMPLIED  -- texture tile for document background --
        %body-color-attrs;  -- bgcolor, text, link, vlink, alink --
        >
```

Example:

```
<body bgcolor=white text=black link=red vlink=maroon alink=fuschia>
```

bgcolor	Specifies the background color for the document body. See following for the syntax of color values.
text	Specifies the color used to stroke the document's text. This is generally used when you have changed the background color with the BGCOLOR or BACKGROUND attributes.
link	Specifies the color used to stroke the text for unvisited hypertext links.
vlink	Specifies the color used to stroke the text for visited hypertext links.
alink	Specifies the highlight color used to stroke the text for hypertext links at the moment the user clicks on the link.
background	Specifies a URL for an image that will be used to tile the document background.

Colors are given in the sRGB color space as hexadecimal numbers (for example, COLOR="#C0FFC0"), or as one of 16 widely understood color names. These colors were originally picked as being the standard 16 colors supported with the Windows VGA palette.

Color Names and sRGB Values

Black = "#000000"	Green = "#008000"
Silver = "#C0C0C0"	Lime = "#00FF00"
Gray = "#808080"	Olive = "#808000"
White = "#FFFFFF"	Yellow = "#FFFF00"
Maroon = "#800000"	Navy = "#000080"
Red = "#FF0000"	Blue = "#0000FF"
Purple = "#800080"	Teal = "#008080"
Fuchsia = "#FF00FF"	Aqua = "#00FFFF"

Block and Text-Level Elements

Most elements that can appear in the document body fall into one of two groups: block-level elements which cause paragraph breaks, and text-level elements which don't. Common block-level elements include H1 to H6 (headers), P (paragraphs) LI (list items), and HR (horizontal rules). Common text-level elements include EM, I, B and FONT (character emphasis), A (hypertext links), IMG and APPLET (embedded objects) and BR (line breaks). Note that block elements generally act as containers for text-level and other block-level elements (excluding headings and address elements), while text-level elements can only contain other text-level elements. The exact model depends on the element.

Headings

```
<!--
  There are six levels of headers from H1 (the most important)
  to H6 (the least important).
-->
<!ELEMENT ( %heading )  - -  (%text;)*>
<!ATTLIST ( %heading )
        align  (left¦center¦right) #IMPLIED
        >
```

H1, H2, H3, H4, H5 and H6 are used for document headings. You always need the start and end tags. H1 elements are more important than H2 elements and so on, so that H6 elements define the least important level of headings. More important headings are generally rendered in a larger font than less important ones. Use the optional ALIGN attribute to set the text alignment within a heading, for example,

```
<H1 ALIGN=CENTER>... centered heading ...</H1>
```

The default is left alignment, but this can be overridden by an enclosing DIV or CENTER element.

ADDRESS

```
<!ENTITY % address.content "((%text;) ¦ P)*">
<!ELEMENT ADDRESS - - %address.content>
```

The ADDRESS element requires start and end tags and specifies information such as authorship and contact details for the current document. User agents should render the content with paragraph-breaks before and after. Note that the content is restricted to paragraphs, plain text, and text-like elements as defined by the %text entity.

Example:

```
<ADDRESS>
Newsletter editor<BR>
J.R. Brown<BR>
8723 Buena Vista, Smallville, CT 01234<BR>
Tel: +1 (123) 456 7890
</ADDRESS>
```


IPR2024-00143 Apple EX1011 Page 471

Block Elements

- **P** *paragraphs:* The paragraph element requires a start tag, but the end tag can always be omitted. Use the ALIGN attribute to set the text alignment within a paragraph, for example, <P ALIGN=RIGHT>
- **UL** *unordered lists:* These require start and end tags, and contain one or more LI elements representing individual list items.
- **OL** *ordered (numbered) lists:* These require start and end tags, and contain one or more LI elements representing individual list items.
- **DL** *definition lists:* These require start and end tags and contain DT elements that give the terms, and DD elements that give corresponding definitions.
- **PRE** *preformatted text:* Requires start and end tags. These elements are rendered with a monospaced font and preserve layout defined by whitespace and line break characters.
- **DIV** *document divisions:* Requires start and end tags. It is used with the ALIGN attribute to set the text alignment of the block elements it contains. ALIGN can be one of LEFT, CENTER, or RIGHT.
- **CENTER** *text alignment:* Requires start and end tags. It is used to center text lines enclosed by the CENTER element. See DIV for a more general solution.
- **BLOCKQUOTE** *quoted passage:* Requires start and end tags. It is used to enclose extended quotations and is typically rendered with indented margins.
- **FORM** *fill-out forms:* Requires start and end tags. This element is used to define a fill-out form for processing by HTTP servers. The attributes are ACTION, METHOD, and ENCTYPE. Form elements can't be nested.
- **ISINDEX** *primitive HTML forms:* Not a container, so the end tag is forbidden. This predates FORM and is used for simple kinds of forms which have a single text input field, implied by this element. A single ISINDEX can appear in the document head or body.
- **HR** *horizontal rules:* Not a container, so the end tag is forbidden. Attributes are ALIGN, NOSHADE, SIZE, and WIDTH.
- **TABLE** *can be nested:* Requires start and end tags. Each table starts with an optional CAPTION followed by one or more TR elements defining table rows. Each row has one or more cells defined by TH or TD elements. Attributes for TABLE elements are WIDTH, BORDER, CELLSPACING, and CELLPADDING.

Paragraphs

```
<!ELEMENT P      - O (%text)*>
<!ATTLIST P
        align  (left¦center¦right) #IMPLIED
        >
```


The P element is used to mark up paragraphs. It is a container and requires a start tag. The end tag is optional as it can always be inferred by the parser. User agents should place paragraph breaks before and after P elements. The rendering is user agent-dependent, but text is generally wrapped to fit the space available.

Example:

```
<P>This is the first paragraph.
<P>This is the second paragraph.
```

Paragraphs are usually rendered flush left with a ragged right margin. The ALIGN attribute can be used to explicitly specify the horizontal alignment:

align=left	The paragraph is rendered flush left.
align=center	The paragraph is centered.
align=right	The paragraph is rendered flush right.

For example:

```
<p align=center>This is a centered paragraph.
<p align=right>and this is a flush right paragraph.
```

The default is left alignment, but this can be overridden by an enclosing DIV or CENTER element.

Lists

List items can contain block-level and text-level items, including nested lists, although headings and address elements are excluded. This limitation is defined via the %flow entity.

Unordered Lists

```
<!ELEMENT UL - -  (LI)+>
<!ENTITY % ULStyle "disc¦square¦circle">
<!ATTLIST UL -- unordered lists --
        type    (%ULStyle)   #IMPLIED   -- bullet style --
        compact (compact)    #IMPLIED   -- reduced interitem spacing --
        >
<!ELEMENT LI - O %flow -- list item -->
<!ATTLIST LI
        type   (%LIStyle)    #IMPLIED   -- list item style --
        >
```

Unordered lists take the form:

```
<UL>
      <LI>... first list item
      <LI>... second list item
      ...
    </UL>
```


IPR2024-00143
Apple EX1011 Page 472

The UL element is used for unordered lists. Both start and end tags are always needed. The LI element is used for individual list items. The end tag for LI elements can always be omitted. Note that LI elements can contain nested lists. The COMPACT attribute can be used as a hint to the user agent to render lists in a more compact style.

The TYPE attribute can be used to set the bullet style on UL and LI elements. The permitted values are disc, square, or circle. The default generally depends on the level of nesting for lists.

with `<li type=disc>`

with `<li type=square>`

with `<li type=circle>`

This list was chosen to cater for the original bullet shapes used by Mosaic in 1993.

Ordered (Numbered) Lists

```
<!ELEMENT OL - -  (LI)+>
<!ATTLIST OL -- ordered lists --
        type      CDATA      #IMPLIED   -- numbering style --
        start     NUMBER     #IMPLIED   -- starting sequence number --
        compact  (compact)   #IMPLIED   -- reduced interitem spacing --
        >

<!ELEMENT LI - O %flow -- list item -->
<!ATTLIST LI
        type      CDATA      #IMPLIED   -- list item style --
        value     NUMBER     #IMPLIED   -- set sequence number --
        >
```

Ordered (or, *numbered*) lists take the form:

```
<OL>
      <LI>... first list item
      <LI>... second list item
      ...
    </OL>
```

The OL START attribute can be used to initialize the sequence number (by default it is initialized to 1). You can set it later on with the VALUE attribute on LI elements. Both of these attributes expect integer values. You can't indicate that numbering should be continued from a previous list, or to skip missing values without giving an explicit number.

The COMPACT attribute can be used as a hint to the user agent to render lists in a more compact style. The OL TYPE attribute allows you to set the numbering style for list items:

Type	*Numbering style*
1	Arabic numbers: 1, 2, 3, ...
a	Lower alpha: a, b, c, ...
A	Upper alpha: A, B, C, ...
i	Lower Roman: i, ii, iii, ...
I	Upper Roman: I, II, III, ...

Definition Lists

```
<!-- definition lists - DT for term, DD for its definition -->

<!ELEMENT DL    - -  (DT|DD)+>
<!ATTLIST DL
        compact (compact) #IMPLIED -- more compact style --
        >

<!ELEMENT DT - O  (%text)*>
<!ELEMENT DD - O  %flow;>
```

Definition lists take the form:

```
<DL>
    <DT> term name
     term definition
    ...
  </DL>
```

DT elements can only act as containers for text-level elements, while DD elements can hold block-level elements as well, excluding headings and address elements.

For example:

```
<DL>
<DT>Term 1This is the definition of the first term.
<DT>Term 2This is the definition of the second term.
</DL>
```

which could be rendered as:

Term 1	This is the definition of the first term.
Term 2	This is the definition of the second term.

The COMPACT attribute can be used with the DL element as a hint to the user agent to render lists in a more compact style.

IPR2024-00143
Apple EX1011 Page 473

DIR and MENU

```
<!ELEMENT (DIR¦MENU) - -  (LI)+ -(%block)>
<!ATTLIST (DIR¦MENU)
        compact (compact) #IMPLIED
        >
```

These elements have been part of HTML from the early days. They are intended for unordered lists similar to UL elements. User agents are recommended to render DIR elements as multicolumn directory lists, and MENU elements as single column menu lists. In practice, Mosaic and most other user agents have ignored this advice and instead render DIR and MENU in an identical way to UL elements.

Preformatted Text

```
<!ELEMENT PRE - - (%text)* -(%pre.exclusion)>
<!ATTLIST PRE
        width NUMBER #implied
        >
```

The PRE element can be used to include preformatted text. User agents render this in a fixed pitch font, preserving spacing associated with white space characters such as space and newline characters. Automatic word-wrap should be disabled within PRE elements.

Note that the SGML standard requires that the parser remove a newline immediately following the start tag or immediately preceding the end tag.

PRE has the same content model as paragraphs, excluding images and elements that produce changes in font size, for example, IMG, BIG, SMALL, SUB, SUP, and FONT.

A few user agents support the WIDTH attribute. It provides a hint to the user agent of the required width in characters. The user agent can use this to select an appropriate font size or to indent the content appropriately.

Here is an example of a PRE element—a verse from Shelley ("To a Skylark"):

```
<PRE>
       Higher still and higher
         From the earth thou springest
       Like a cloud of fire;
         The blue deep thou wingest,
And singing still dost soar, and soaring ever singest.
</PRE>
```

which is rendered as:

```
       Higher still and higher
         From the earth thou springest
       Like a cloud of fire;
         The blue deep thou wingest,
And singing still dost soar, and soaring ever singest.
```


The horizontal tab character (encoded in Unicode, US ASCII, and ISO 8859-1 as decimal 9) should be interpreted as the smallest nonzero number of spaces which will leave the number of characters so far on the line as a multiple of 8. Its use is strongly discouraged since it is common practice when editing to set the tab-spacing to other values, leading to misaligned documents.

XMP, LISTING, and PLAINTEXT

```
<![ %HTML.Deprecated [

<!ENTITY % literal "CDATA"
        -- historical, non-conforming parsing mode where
           the only markup signal is the end tag
           in full
        -->

<!ELEMENT (XMP¦LISTING) - -  %literal>
<!ELEMENT PLAINTEXT - O %literal>

]]>
```

These are obsolete tags for preformatted text that predate the introduction of PRE. User agents may support these for backwards compatibility. Authors should avoid using them in new documents!

DIV and CENTER

```
<!ELEMENT DIV - - %body.content>
<!ATTLIST DIV
        align   (left¦center¦right) #IMPLIED -- alignment of following text --
        >

<!-- CENTER is a shorthand for DIV with ALIGN=CENTER -->
<!ELEMENT center - - %body.content>
```

DIV elements can be used to structure HTML documents as a hierarchy of divisions. The ALIGN attribute can be used to set the default horizontal alignment for elements within the content of the DIV element. Its value is restricted to LEFT, CENTER, or RIGHT, and is defined in the same way as for the paragraph element <P>.

Note that because DIV is a block-like element, it will terminate an open P element. Other than this, user agents are **not** expected to render paragraph breaks before and after DIV elements. CENTER is directly equivalent to DIV with ALIGN=CENTER. Both DIV and CENTER require start and end tags.

CENTER was introduced by Netscape before they added support for the HTML 3.0 DIV element. It is retained in HTML 3.2 on account of its widespread deployment.

IPR2024-00143
Apple EX1011 Page 474

BLOCKQUOTE

```
<!ELEMENT BLOCKQUOTE - - %body.content>
```

This is used to enclose block quotations from other works. Both the start and end tags are required. It is often rendered indented, for example:

> They went in single file, running like hounds on a strong scent, and an eager light was in their eyes. Nearly due west the broad swath of the marching Orcs tramped its ugly slot; the sweet grass of Rohan had been bruised and blackened as they passed.
>
> from "The Two Towers" by J.R.R. Tolkien.

FORM

```
<!ENTITY % HTTP-Method "GET ¦ POST"
        -- as per HTTP specification
        -->

<!ELEMENT FORM - - %body.content -(FORM)>
<!ATTLIST FORM
        action %URL #IMPLIED  -- server-side form handler --
        method (%HTTP-Method) GET -- see HTTP specification --
        enctype %Content-Type; "application/x-www-form-urlencoded"
        >
```

This is used to define an HTML form, and you can have more than one form in the same document. Both the start and end tags are required. For very simple forms, you can also use the `ISINDEX` element. Forms can contain a wide range of HTML markup, including several kinds of form fields such as single- and multiline text fields, radio button groups, checkboxes, and menus.

`action`	This specifies a URL which is either used to post forms via e-mail, for example `action="mailto:foo@bar.com"`, or used to invoke a server-side forms handler via HTTP, for example `action="http://www.acme.com/cgi-bin/register.pl"`.
`method`	When the action attribute specifies an HTTP server, the method attribute determines which HTTP method will be used to send the form's contents to the server. It can be either `GET` or `POST`, and defaults to `GET`.
`enctype`	This determines the mechanism used to encode the form's contents. It defaults to `application/x-www-form-urlencoded`.

Further details on handling forms are given in RFC 1867.

HR—Horizontal Rules

Horizontal rules may be used to indicate a change in topic. In a speech-based user agent, the rule could be rendered as a pause.


```
<!ELEMENT HR    - O EMPTY>
<!ATTLIST HR
        align (left¦right¦center) #IMPLIED
        noshade (noshade) #IMPLIED
        size  %Pixels #IMPLIED
        width %Length #IMPLIED
        >
```

`HR` elements are not containers so the end tag is forbidden. The attributes are `ALIGN`, `NOSHADE`, `SIZE`, and `WIDTH`.

`align`	This determines whether the rule is placed at the left, center or right of the space between the current left and right margins for `align=left`, `align=center`, or `align=right`, respectively. By default, the rule is centered.
`noshade`	This attribute requests the user agent to render the rule in a solid color rather than as the traditional two color "groove."
`size`	This can be used to set the height of the rule in pixels.
`width`	This can be used to set the width of the rule in pixels (for example, `width=100`) or as the percentage between the current left and right margins (for example, `width="50%"`). The default is 100%.

B HTML 3.2 REFERENCE SPECIFICATION

Tables

HTML 3.2 includes a widely deployed subset of the specification given in RFC 1942 and can be used to mark up tabular material or for layout purposes. Note that the latter role typically causes problems when rendering to speech or to text-only user agents.

```
<!-- horizontal placement of table relative to window -->
<!ENTITY % Where "(left¦center¦right)">

<!-- horizontal alignment attributes for cell contents -->
<!ENTITY % cell.halign
        "align  (left¦center¦right) #IMPLIED"
        >

<!-- vertical alignment attributes for cell contents -->
<!ENTITY % cell.valign
        "valign  (top¦middle¦bottom)  #IMPLIED"
        >

<!ELEMENT table - - (caption?, tr+)>
<!ELEMENT tr - O (th¦td)*>
<!ELEMENT (th¦td) - O %body.content>

<!ATTLIST table                          -- table element --
        align     %Where;   #IMPLIED  -- table position relative to window --
        width     %Length   #IMPLIED  -- table width relative to window --
        border    %Pixels   #IMPLIED  -- controls frame width around table --
        cellspacing %Pixels #IMPLIED  -- spacing between cells --
        cellpadding %Pixels #IMPLIED  -- spacing within cells --
        >
```

IPR2024-00143
Apple EX1011 Page 475

```
<!ELEMENT CAPTION - - (%text;)* -- table or figure caption -->
<!ATTLIST CAPTION
        align (top|bottom) #IMPLIED
        >

<!ATTLIST tr                        -- table row --
        %cell.halign;               -- horizontal alignment in cells --
        %cell.valign;               -- vertical alignment in cells --
        >

<!ATTLIST (th|td)                   -- header or data cell --
        nowrap (nowrap)  #IMPLIED   -- suppress word wrap --
        rowspan NUMBER   1          -- number of rows spanned by cell --
        colspan NUMBER   1          -- number of cols spanned by cell --
        %cell.halign;               -- horizontal alignment in cells --
        %cell.valign;               -- vertical alignment in cells --
        width   %Pixels  #IMPLIED   -- suggested width for cell --
        height  %Pixels  #IMPLIED   -- suggested height for cell --
        >
```

Tables take the general form:

```
<TABLE BORDER=3 CELLSPACING=2 CELLPADDING=2 WIDTH="80%">
<CAPTION> ... table caption ... </CAPTION>
<TR><TD> first cell <TD> second cell
<TR> ...
...
</TABLE>
```

The attributes on TABLE are all optional. By default, the table is rendered without a surrounding border. The table is generally sized automatically to fit the contents, but you can also set the table width using the WIDTH attribute. BORDER, CELLSPACING, and CELLPADDING provide further control over the table's appearance. Captions are rendered at the top or bottom of the table depending on the ALIGN attribute.

Each table row is contained in a TR element, although the end tag can always be omitted. Table cells are defined by TD elements for data and TH elements for headers. Like TR, these are containers and can be given without trailing end tags. TH and TD support several attributes: ALIGN and VALIGN for aligning cell content, ROWSPAN and COLSPAN for cells which span more than one row or column. A cell can contain a wide variety of other block- and text-level elements, including form fields and other tables.

The TABLE element always requires both start and end tags. It supports the following attributes:

align — This takes one of the case insensitive values: LEFT, CENTER, or RIGHT. It specifies the horizontal placement of the table relative to the current left and right margins. It defaults to left alignment, but this can be overridden by an enclosing DIV or CENTER element.

width — In the absence of this attribute, the table width is automatically determined from the table contents. You can use the WIDTH attribute to set the table width to a fixed value in pixels (for example WIDTH=212) or as a percentage of the space between the current left and right margins (for example WIDTH="80%").

border — This attribute can be used to specify the width of the outer border around the table to a given number of pixels (for example BORDER=4). The value can be set to zero to suppress the border altogether. In the absence of this attribute, the border should be suppressed. Note that some browsers also accept <TABLE BORDER> with the same semantics as BORDER=1.

cellspacing — In traditional desktop publishing software, adjacent table cells share a common border. This is not the case in HTML. Each cell is given its own border which is separated from the borders around neighboring cells. This separation can be set in pixels using the CELLSPACING attribute, (for example CELLSPACING=10). The same value also determines the separation between the table border and the borders of the outermost cells.

cellpadding — This sets the padding in pixels between the border around each cell and the cell's contents.

The CAPTION element has one attribute, ALIGN, which can be either ALIGN=TOP or ALIGN=BOTTOM. This can be used to force the caption to be placed above the top or below the bottom of the table, respectively. Most user agents default to placing the caption above the table. CAPTION always requires both start and end tags. Captions are limited to plain text and text-level elements as defined by the %text entity. Block-level elements are not permitted.

The TR or table row element requires a start tag, but the end tag can always be left out. TR acts as a container for table cells. It has two attributes:

align — Sets the default horizontal alignment of cell contents. It takes one of the case insensitive values—LEFT, CENTER, or RIGHT—and plays the same role as the ALIGN attribute on paragraph elements.

valign — This can be used to set the default vertical alignment of cell contents within each cell. It takes one of the case-insensitive values—TOP, MIDDLE, or BOTTOM—to position the cell contents at the top, middle, or bottom of the cell, respectively.

There are two elements for defining table cells. TH is used for header cells and TD for data cells. This distinction allows user agents to render header and data cells in different fonts, and enables speech based browsers to do a better job. The start tags for TH and TD are always needed, but the end tags can be left out. Table cells can have the following attributes:

IPR2024-00143
Apple EX1011 Page 476

nowrap	The presence of this attribute disables automatic word wrap within the contents of this cell (for example <TD NOWRAP>). This is equivalent to using the entity for nonbreaking spaces within the content of the cell.
rowspan	This takes a positive integer value specifying the number of rows spanned by this cell. It defaults to one.
colspan	This takes a positive integer value specifying the number of columns spanned by this cell. It defaults to one.
align	Specifies the default horizontal alignment of cell contents, and overrides the ALIGN attribute on the table row. It takes the same values: LEFT, CENTER, and RIGHT. If you don't specify an ALIGN attribute value on the cell, the default is left alignment for <td> and center alignment for <th>, although you can override this with an ALIGN attribute on the TR element.
valign	Specifies the default vertical alignment of cell contents, overriding the VALIGN attribute on the table row. It takes the same values: TOP, MIDDLE, and BOTTOM. If you don't specify a VALIGN attribute value on the cell, the default is middle although you can override this with a VALIGN attribute on the TR element.
width	Specifies the suggested width for a cell content in pixels excluding the cell padding. This value will normally be used except when it conflicts with the width requirements for other cells in the same column.
height	Specifies the suggested height for a cell content in pixels excluding the cell padding. This value will normally be used except when it conflicts with the height requirements for other cells in the same row.

Tables are commonly rendered in bas-relief, raised up with the outer border as a bevel, and individual cells inset into this raised surface. Borders around individual cells are only drawn if the cell has explicit content. White space doesn't count for this purpose with the exception of .

The algorithms used to automatically size tables should take into account the minimum and maximum width requirements for each cell. This is used to determine the minimum and maximum width requirements for each column and hence for the table itself.

Cells spanning more than one column contribute to the widths of each of the columns spanned. One approach is to evenly apportion the cell's minimum and maximum width between these columns; another is to weight the apportioning according to the contributions from cells that don't span multiple columns.

For some user agents it may be necessary or desirable to break text lines within words. In such cases a visual indication that this has occurred is advised.

The minimum and maximum width of nested tables contribute to the minimum and maximum width of the cell in which they occur. Once the width requirements are known for the top-level table, the column widths for that table can be assigned. This allows the widths of nested tables to be assigned and hence, in turn, the column widths of such tables. If practical, all columns should be assigned at least their minimum widths. It is suggested that any surplus space is then shared out proportional to the difference between the minimum and maximum width requirements of each column.

Note that pixel values for width and height refer to screen pixels, and should be multiplied by an appropriate factor when rendering to very high resolution devices such as laser printers. For instance, if a user agent has a display with 75 pixels per inch and is rendering to a laser printer with 600 dots per inch, then the pixel values given in HTML attributes should be multiplied by a factor of 8.

Text-Level Elements

These don't cause paragraph breaks. Text-level elements that define character styles can generally be nested. They can contain other text-level elements but not block-level elements.

- Font style elements
- Phrase elements
- Form fields
- The A (anchor) element
- IMG—inline images
- APPLET *(Java Applets)*
- FONT elements
- BASEFONT elements
- BR—line breaks
- MAP—client-side image maps

Font Style Elements

These all require start and end tags, for example:

```
This has some <B>bold text.</B>
```

Text-level elements must be properly nested; the following is in error:

```
This has some <B>bold and <I></B>italic text</I>.
```

User agents should do their best to respect nested emphasis, for example:

```
This has some <B>bold and <I>italic text</I></B>.
```


IPR2024-00143
Apple EX1011 Page 477

Where the available fonts are restricted or for speech output, alternative means should be used for rendering differences in emphasis.

TT	Teletype or monospaced text
I	Italic text style
B	Bold text style
U	Underlined text style
STRIKE	Strike-through text style
BIG	Places text in a large font
SMALL	Places text in a small font
SUB	Places text in subscript style
SUP	Places text in superscript style

Note: future revisions to HTML may phase out STRIKE *in favor of the more concise* S *tag from HTML 3.0.*

Phrase Elements

These all require start and end tags, for example:

```
This has some <EM>emphasized text</EM>.
```

EM	Basic emphasis typically rendered in an italic font
STRONG	Strong emphasis typically rendered in a bold font
DFN	Defining instance of the enclosed term
CODE	Used for extracts from program code
SAMP	Used for sample output from programs, scripts, etc.
KBD	Used for text to be typed by the user
VAR	Used for variables or arguments to commands
CITE	Used for citations or references to other sources

Form Fields

INPUT, SELECT, and TEXTAREA are only allowed within FORM elements. INPUT can be used for a variety of form fields, including single-line text fields, password fields, checkboxes, radio buttons, submit and reset buttons, hidden fields, file upload, and image buttons. SELECT elements are used for single or multiple choice menus. TEXTAREA elements are used to define multiline text fields. The content of the element is used to initialize the field.

INPUT Text Fields, Radio Buttons, Checkboxes...

INPUT elements are not containers, so the end tag is forbidden.

```
<!ENTITY % IAlign "(top¦middle¦bottom¦left¦right)">

<!ENTITY % InputType
        "(TEXT ¦ PASSWORD ¦ CHECKBOX ¦ RADIO ¦ SUBMIT
            ¦ RESET ¦ FILE ¦ HIDDEN ¦ IMAGE)">

<!ELEMENT INPUT - O EMPTY>
<!ATTLIST INPUT
        type %InputType TEXT      -- what kind of widget is needed --
        name  CDATA #IMPLIED      -- required for all but submit and reset --
        value CDATA #IMPLIED      -- required for radio and checkboxes --
        checked (checked) #IMPLIED -- for radio buttons and check boxes --
        size CDATA  #IMPLIED      -- specific to each type of field --
        maxlength NUMBER #IMPLIED
        src   %URL  #IMPLIED      -- for fields with background images --
        align %IAlign #IMPLIED    -- vertical or horizontal alignment --
        >
```

type

Used to set the type of input field, as follows:

type=text (the default)

A single-line text field whose visible size can be set using the size attribute, for example size=40 for a 40 character wide field. Users should be able to type more than this limit, though, with the text scrolling through the field to keep the input cursor in view. You can enforce an upper limit on the number of characters that can be entered with the MAXLENGTH attribute. The NAME attribute is used to name the field, while the VALUE attribute can be used to initialize the text string shown in the field when the document is first loaded.

```
<input type=text size=40 name=user value="your name">
```

type=password

This is like type=text, but echoes characters using a character like * to hide the text from prying eyes when entering passwords. You can use SIZE and MAXLENGTH attributes to control the visible and maximum length exactly as per regular text fields.

```
<input type=password size=12 name=pw>
```

type=checkbox

Used for simple Boolean attributes, or for attributes that can take multiple values at the same time. The latter is represented by several checkbox fields with the same name

IPR2024-00143
Apple EX1011 Page 478

and a different value attribute. Each checked checkbox generates a separate name/value pair in the submitted data, even if this results in duplicate names. Use the checked attribute to initialize the checkbox to its checked state.

```
<input type=checkbox checked name=uscitizen value=yes>
```

type=radio

Used for attributes which can take a single value from a set of alternatives. Each radio button field in the group should be given the same name. Radio buttons require an explicit value attribute. Only the checked radio button in the group generates a name/value pair in the submitted data. One radio button in each group should be initially checked using the checked attribute.

```
<input type=radio name=age value="0-12">
<input type=radio name=age value="13-17">
<input type=radio name=age value="18-25">
<input type=radio name=age value="26-35" checked>
<input type=radio name=age value="36-">
```

type=submit

This defines a button that users can click to submit the form's contents to the server. The button's label is set from the `VALUE` attribute. If the `NAME` attribute is given, then the submit button's name/value pair will be included in the submitted data. You can include several submit buttons in the form. See `type=image` for graphical submit buttons.

```
<input type=submit value="Party on ...">
```

type=image

This is used for graphical submit buttons rendered by an image rather than a text string. The URL for the image is specified with the `SRC` attribute. The image alignment can be specified with the `ALIGN` attribute. In this respect, graphical submit buttons are treated identically to `IMG` elements, so you can set align to left, right, top, middle, or bottom. The x and y values of the location clicked are passed to the server: In the submitted data, image fields are included as two name/value pairs. The names are derived by taking the name of the field and appending ".x" for the x value, and ".y" for the y value.

```
<p>Now choose a point on the map:

    <input type=image name=point src="map.gif">
```

Note: image fields typically cause problems for text-only and speech-based user agents!

type=reset

This defines a button that users can click to reset form fields to their initial state when the document was first loaded. You can set the label by providing a `value` attribute. Reset buttons are never sent as part of the form's contents.

```
<input type=reset value="Start over ...">
```


type=file

This provides a means for users to attach a file to the form's contents. It is generally rendered by text field and an associated button which when clicked invokes a file browser to select a filename. The filename can also be entered directly in the text field. Just like `type=text`, you can use the `SIZE` attribute to set the visible width of this field in average character widths. You can set an upper limit to the length of filenames using the `MAXLENGTH` attribute. Some user agents support the ability to restrict the kinds of files to those matching a comma-separated list of MIME content types given with the `ACCEPT` attribute, for example `accept="image/*"` restricts files to images. Further information can be found in RFC 1867.

```
<input type=file name=photo size=20 accept="image/*">
```

type=hidden

These fields should not be rendered and provide a means for servers to store state information with a form. This will be passed back to the server when the form is submitted, using the name/value pair defined by the corresponding attributes. This is a work around for the statefulness of HTTP. Another approach is to use HTTP "Cookies."

```
<input type=hidden name=customerid value="c2415-345-8563">
```

name

Used to define the property name that will be used to identify this field's content when it is submitted to the server.

value

Used to initialize the field, or to provide a textual label for submit and reset buttons.

checked

The presence of this attribute is used to initialize checkboxes and radio buttons to their checked state.

size

Used to set the visible size of text fields to a given number of average character widths, for example `size=20`.

maxlength

Sets the maximum number of characters permitted in a text field.

src

Specifies a URL for the image to use with a graphical submit button.

IPR2024-00143
Apple EX1011 Page 479

align

Used to specify image alignment for graphical submit buttons. It is defined just like the IMG align attribute and takes one of the values top, middle, bottom, left or right, defaulting to bottom.

SELECT Menus

```
<!ELEMENT SELECT - - (OPTION+)>
<!ATTLIST SELECT
        name CDATA #REQUIRED
        size NUMBER #IMPLIED
        multiple (multiple) #IMPLIED
        >

<!ELEMENT OPTION - O (#PCDATA)*>
<!ATTLIST OPTION
        selected (selected) #IMPLIED
        value  CDATA  #IMPLIED -- defaults to element content --
        >
```

SELECT is used to define select one from many or many from many menus. SELECT elements require start and end tags and contain one or more OPTION elements that define menu items. One from many menus are generally rendered as drop-down menus while many from many menus are generally shown as list boxes.

Example:

```
<SELECT NAME="flavor">
<OPTION VALUE=a>Vanilla
<OPTION VALUE=b>Strawberry
<OPTION VALUE=c>Rum and Raisin
<OPTION VALUE=d>Peach and Orange
</SELECT>
```

SELECT attributes:

- **name** — This specifies a property name that is used to identify the menu choice when the form is submitted to the server. Each selected option results in a property name/value pair being included as part of the form's contents.
- **size** — This sets the number of visible choices for many menus.
- **multiple** — The presence of this attribute signifies that the users can make multiple selections. By default only one selection is allowed.

OPTION attributes:

- **selected** — When this attribute is present, the option is selected when the document is initially loaded. It is an error for more than one option to be so selected for one from many menus.

- **value** — Specifies the property value to be used when submitting the form's content. This is combined with the property name as given by the NAME attribute of the parent SELECT element.

TEXTAREA Multiline Text Fields

```
<!-- Multi-line text input field. -->

<!ELEMENT TEXTAREA - - (#PCDATA)*>
<!ATTLIST TEXTAREA
        name CDATA #REQUIRED
        rows NUMBER #REQUIRED
        cols NUMBER #REQUIRED
        >
```

TEXTAREA elements require start and end tags. The content of the element is restricted to text and character entities. It is used to initialize the text that is shown when the document is first loaded.

Example:

```
<TEXTAREA NAME=address ROWS=4 COLS=40>
Your address here ...
</TEXTAREA>
```

It is recommended that user agents canonize line endings to CR, LF (ASCII decimal 13, 10) when submitting the field's contents. The character set for submitted data should be ISO Latin-1, unless the server has previously indicated that it can support alternative character sets.

- **name** — This specifies a property name that is used to identify the textarea field when the form is submitted to the server.
- **rows** — Specifies the number of visible text lines. Users should be able to enter more lines that this, so user agents should provide some means to scroll through the contents of the textarea field when the contents extend beyond the visible area.
- **cols** — Specifies the visible width in average character widths. Users should be able to enter longer lines that this, so user agents should provide some means to scroll through the contents of the textarea field when the contents extend beyond the visible area. User agents may wrap visible text lines to keep long lines visible without the need for scrolling.

Special Text-Level Elements

A (anchor), IMG, APPLET, FONT, BASEFONT, BR, and MAP.

IPR2024-00143
Apple EX1011 Page 480

The A (anchor) Element

```
<!ELEMENT A - - (%text)* -(A)>
<!ATTLIST A
        name    CDATA   #IMPLIED    -- named link end --
        href    %URL    #IMPLIED    -- URL for linked resource --
        rel     CDATA   #IMPLIED    -- forward link types --
        rev     CDATA   #IMPLIED    -- reverse link types --
        title   CDATA   #IMPLIED    -- advisory title string --
        >
```

Anchors can't be nested and always require start and end tags. They are used to define hypertext links and also to define named locations for use as targets for hypertext links, for example:

```
The way to <a href="hands-on.html">happiness</a>.
```

They are also used to define named locations for use as targets for hypertext links, for example:

```
<h2><a name=mit>545 Tech Square - Hacker's Paradise</a></h2>
```

`name` This should be a string defining unique name for the scope of the current HTML document. `NAME` is used to associate a name with this part of a document for use with URLs that target a named section of a document.

`href` Specifies a URL acting as a network address for the linked resource. This could be another HTML document, a PDF file, an image, and so on.

`rel` The forward relationship also known as the "link type". It can be used to determine to how deal with the linked resource when printing out a collection of linked resources.

`rev` This defines a reverse relationship. A link from document A to document B with `REV=relation` expresses the same relationship as a link from B to A with `REL=relation`. `REV=made` is sometimes used to identify the document author, either the author's e-mail address with a mailto URL, or a link to the author's home page.

`title` An advisory title for the linked resource.

IMG—Inline Images

```
<!ENTITY % IAlign "(top|middle|bottom|left|right)">

<!ELEMENT IMG    - O EMPTY --  Embedded image -->
<!ATTLIST IMG
        src     %URL     #REQUIRED  -- URL of image to embed --
        alt     CDATA    #IMPLIED   -- for display in place of image --
        align   %IAlign  #IMPLIED   -- vertical or horizontal alignment --
        height  %Pixels  #IMPLIED   -- suggested height in pixels --
        width   %Pixels  #IMPLIED   -- suggested width in pixels --
```



```
        border  %Pixels  #IMPLIED   -- suggested link border width --
        hspace  %Pixels  #IMPLIED   -- suggested horizontal gutter --
        vspace  %Pixels  #IMPLIED   -- suggested vertical gutter --
        usemap  %URL     #IMPLIED   -- use client-side image map --
        ismap   (ismap)  #IMPLIED   -- use server image map --
        >
```

Used to insert images. `IMG` is an empty element and so the end tag is forbidden. Images can be positioned vertically relative to the current textline or floated to the left or right. See `BR` with the `CLEAR` attribute for control over textflow.

```
e.g. <IMG SRC="canyon.gif" ALT="Grand Canyon">
```

`IMG` elements support the following attributes:

`src` This attribute is required for every `IMG` element. It specifies a URL for the image resource, for instance a GIF, JPEG or PNG image file.

`alt` This is used to provide a text description of the image and is vital for interoperability with speech-based and text only user agents.

`align` This specifies how the image is positioned relative to the current textline in which it occurs:

`align=top` positions the top of the image with the top of the current text line. User agents vary in how they interpret this. Some only take into account what has occurred on the text line prior to the `IMG` element and ignore what happens after it.

`align=middle` aligns the middle of the image with the baseline for the current textline.

`align=bottom` is the default and aligns the bottom of the image with the baseline.

`align=left` floats the image to the current left margin, temporarily changing this margin, so that subsequent text is flowed along the image's righthand side. The rendering depends on whether there is any left aligned text or images that appear earlier than the current image in the markup. Such text (but not images) generally forces left-aligned images to wrap to a new line, with the subsequent text continuing on the former line.

`align=right` floats the image to the current right margin, temporarily changing this margin, so that subsequent text is flowed along the image's lefthand side. The rendering depends on whether there is any right aligned text or images that appear earlier than the current image in the markup. Such text (but not images) generally forces right-aligned images to wrap to a new line, with the subsequent text continuing on the former line.

IPR2024-00143
Apple EX1011 Page 481

Note that some browsers introduce spurious spacing with multiple left- right-aligned images. As a result, authors can't depend on this being the same for browsers from different vendors. See BR for ways to control text flow.

width	Specifies the intended width of the image in pixels. When given together with the height, this allows user agents to reserve screen space for the image before the image data has arrived over the network.
height	Specifies the intended height of the image in pixels. When given together with the width, this allows user agents to reserve screen space for the image before the image data has arrived over the network.
border	When the IMG element appears as part of a hypertext link, the user agent will generally indicate this by drawing a colored border (typically blue) around the image. This attribute can be used to set the width of this border in pixels. Use border=0 to suppress the border altogether. User agents are recommended to provide additional cues that the image is clickable, for example by changing the mouse pointer.
hspace	This can be used to provide white space to the immediate left and right of the image. The HSPACE attribute sets the width of this white space in pixels. By default HSPACE is a small, nonzero number.
vspace	This can be used to provide white space above and below the image The VSPACE attribute sets the height of this white space in pixels. By default VSPACE is a small, nonzero number.
usemap	This can be used to give a URL fragment identifier for a client-side image map defined with the MAP element.
ismap	When the IMG element is part of a hypertext link, and the user clicks on the image, the ISMAP attribute causes the location to be passed to the server. This mechanism causes problems for text-only and speech-based user agents. Whenever it's possible to do so, use the MAP element instead.

Here is an example of how you use ISMAP:

```
<a href="/cgibin/navbar.map"><img src=navbar.gif ismap border=0></a>
```

The location clicked is passed to the server as follows. The user agent derives a new URL from the URL specified by the HREF attribute by appending ?, the x coordinate, and the y coordinate of the location in pixels. The link is then followed using the new URL. For instance, if the user

clicked at the location x=10, y=27 then the derived URL will be "/cgibin/navbar.map?10,27". It is generally a good idea to suppress the border and use graphical idioms to indicate that the image is clickable.

Note that pixel values refer to screen pixels, and should be multiplied by an appropriate factor when rendering to very high resolution devices such as laser printers. For instance, if a user agent has a display with 75 pixels per inch and is rendering to a laser printer with 600 dots per inch, then the pixel values given in HTML attributes should be multiplied by a factor of 8.

APPLET (Java Applets)

```
<!ELEMENT APPLET - - (PARAM | %text)*>
<!ATTLIST APPLET
        codebase %URL     #IMPLIED   -- code base --
        code     CDATA    #REQUIRED  -- class file --
        alt      CDATA    #IMPLIED   -- for display in place of applet --
        name     CDATA    #IMPLIED   -- applet name --
        width    %Pixels  #REQUIRED  -- suggested width in pixels --
        height   %Pixels  #REQUIRED  -- suggested height in pixels --
        align    %IAlign  #IMPLIED   -- vertical or horizontal alignment --
        hspace   %Pixels  #IMPLIED   -- suggested horizontal gutter --
        vspace   %Pixels  #IMPLIED   -- suggested vertical gutter --
        >

<!ELEMENT PARAM - O EMPTY>
<!ATTLIST PARAM
    name    NMTOKEN   #REQUIRED  -- The name of the parameter --
    value   CDATA     #IMPLIED   -- The value of the parameter --
    >
```

Requires start and end tags. This element is supported by all Java-enabled browsers. It allows you to embed a Java applet into HTML documents. APPLET uses associated PARAM elements to pass parameters to the applet. Following the PARAM elements, the content of APPLET elements should be used to provide an alternative to the applet for user agents that don't support Java. It is restricted to text-level markup as defined by the %text entity in the DTD. Java-compatible browsers ignore this extra HTML code. You can use it to show a snapshot of the applet running, with text explaining what the applet does. Other possibilities for this area are a link to a page that is more useful for the Java-ignorant browser, or text that taunts the user for not having a Java-compatible browser.

Here is a simple example of a Java applet:

```
<applet code="Bubbles.class" width=500 height=500>
Java applet that draws animated bubbles.
</applet>
```

Here is another one using a PARAM element:

```
<applet code="AudioItem" width=15 height=15>
<param name=snd value="Hello.au¦Welcome.au">
Java applet that plays a welcoming sound.
</applet>
```


IPR2024-00143
Apple EX1011 Page 482

`codebase = codebaseURL`	This optional attribute specifies the base URL of the applet—the directory or folder that contains the applet's code. If this attribute is not specified, then the document's URL is used.
`code = appletFile`	This required attribute gives the name of the file that contains the applet's compiled Applet subclass. This file is relative to the base URL of the applet. It cannot be absolute.
`alt = alternateText`	This optional attribute specifies any text that should be displayed if the browser understands the `<APPLET>` tag but can't run Java applets.
`name = appletInstanceName`	This optional attribute specifies a name for the applet instance, which makes it possible for applets on the same page to find (and communicate with) each other.
`width = pixels` `height = pixels`	These required attributes give the initial width and height (in pixels) of the applet display area, not counting any windows or dialogs that the applet brings up.
`align = alignment`	This attribute specifies the alignment of the applet. This attribute is defined in exactly the same way as the `IMG` element. The permitted values are top, middle, bottom, left, and right. The default is bottom.
`vspace = pixels` `hspace = pixels`	These optional attributes specify the number of pixels above and below the applet (`VSPACE`) and on each side of the applet (`HSPACE`). They're treated the same way as the IMG element's `VSPACE` and `HSPACE` attributes.

The `PARAM` element is used to pass named parameters to applet:

```
<PARAM NAME = appletParameter VALUE = value>
```

`PARAM` elements are the only way to specify applet-specific parameters. Applets read user-specified values for parameters with the `getParameter()` method.

`name = applet parameter name`

`value = parameter value`

SGML character entities such `é`, `"` and `¹` are expanded before the parameter value is passed to the applet. To include an `&` character, use `&`.

Note: `PARAM` elements should be placed at the start of the content for the `APPLET` element. This is not specified as part of the DTD due to technicalities with SGML mixed content models.

FONT

```
<!ELEMENT FONT - - (%text)*      -- local change to font -->
<!ATTLIST FONT
    size    CDATA   #IMPLIED     -- [+]nn e.g. size="+1", size=4 --
    color   CDATA   #IMPLIED     -- #RRGGBB in hex, e.g. red: color="#FF0000" --
    >
```

Requires start and end tags. This allows you to change the font size and/or color for the enclosed text.

The attributes are `SIZE` and `COLOR`. Font sizes are given in terms of a scalar range defined by the user agent with no direct mapping to point sizes etc. The `FONT` element may be phased out in future revisions to HTML.

`size`	This sets the font size for the contents of the font element. You can set size to an integer ranging from 1 to 7 for an absolute font size, or specify a relative font size with a signed integer value, for example `size="+1"` or `size="-2"`. This is mapped to an absolute font size by adding the current base font size as set by the `BASEFONT` element (see following).
`color`	Used to set the color to stroke the text. Colors are given as RGB in hexadecimal notation or as one of 16 widely understood color names defined as per the `BGCOLOR` attribute on the `BODY` element.

Some user agents also support a `FACE` attribute which accepts a comma separated list of font names in order of preference. This is used to search for an installed font with the corresponding name. `FACE` is not part of HTML 3.2.

The following shows the effects of setting font to absolute sizes:

size=1 size=2 size=3 size=4 size=5 size=6 size=7

The following shows the effect of relative font sizes using a base font size of 3:

size=-4 size=-3 size=-2 size=-1 size=+1 size=+2 size=+3 size=+4

The same thing with a base font size of 6:

size=-4 size=-3 size=-2 size=-1 size=+1 size=+2 size=+3 size=+4

IPR2024-00143
Apple EX1011 Page 483

BASEFONT

```
<!ELEMENT BASEFONT - O EMPTY     -- base font size (1 to 7) -->
<!ATTLIST BASEFONT
    size     CDATA    #IMPLIED    -- e.g. size=4, defaults to 3 --
    >
```

Used to set the base font size. BASEFONT is an empty element, so the end tag is forbidden. The SIZE attribute is an integer value ranging from 1 to 7. The base font size applies to the normal and preformatted text but not to headings, except where these are modified using the FONT element with a relative font size.

BR

Used to force a line break. This is an empty element, so the end tag is forbidden. The CLEAR attribute can be used to move down past floating images on either margin. <BR CLEAR=LEFT> moves down past floating images on the left margin, <BR CLEAR=RIGHT> does the same for floating images on the right margin, while <BR CLEAR=ALL> does the same for such images on both left and right margins.

MAP

The MAP element provides a mechanism for client-side image maps. These can be placed in the same document or grouped in a separate document, although this isn't yet widely supported. The MAP element requires start and end tags. It contains one or more AREA elements that specify hotzones on the associated image and bind these hotzones to URLs.

```
<!ENTITY % SHAPE "(rect|circle|poly)">
<!ENTITY % COORDS "CDATA" -- comma separated list of numbers -->

<!ELEMENT MAP - - (AREA)+>
<!ATTLIST MAP
    name     CDATA    #REQUIRED
    >

<!ELEMENT AREA - O EMPTY>
<!ATTLIST AREA
    shape    %SHAPE   rect
    coords   %COORDS  #IMPLIED   -- defines coordinates for shape --
    href     %URL     #IMPLIED   -- this region acts as hypertext link --
    nohref   (nohref) #IMPLIED   -- this region has no action --
    alt      CDATA    #REQUIRED  -- needed for non-graphical user agents --
    >
```

Here is a simple example for a graphical navigational toolbar:

```
<img src="navbar.gif" border=0 usemap="#map1">

<map name="map1">
 <area href=guide.html alt="Access Guide" shape=rect coords="0,0,118,28">
 <area href=search.html alt="Search" shape=rect coords="184,0,276,28">
 <area href=shortcut.html alt="Go" shape=rect coords="118,0,184,28">
 <area href=top10.html alt="Top Ten" shape=rect coords="276,0,373,28">
</map>
```


The MAP element has one attribute, NAME, which is used to associate a name with a map. This is then used by the USEMAP attribute on the IMG element to reference the map via a URL fragment identifier. Note that the value of the NAME attribute is case sensitive.

The AREA element is an empty element, and so the end tag is forbidden. It takes the following attributes: SHAPE, COORDS, HREF, NOHREF, and ALT. The SHAPE and COORDS attributes define a region on the image. If the SHAPE attribute is omitted, SHAPE="RECT" is assumed.

```
shape=rect coords="left-x, top-y, right-x, bottom-y"
shape=circle coords="center-x, center-y, radius"
shape=poly coords="x1,y1,x2,y2,x3,y3, ..."
```

Where x and y are measured in pixels from the left/top of the associated image. If x and y values are given with a percent sign as a suffix, the values should be interpreted as percentages of the image's width and height, respectively. For example:

```
SHAPE=RECT COORDS="0, 0, 50%, 100%"
```

The HREF attribute gives a URL for the target of the hypertext link. The NOHREF attribute is used when you want to define a region that doesn't act as a hotzone. This is useful when you want to cut a hole in an underlying region acting as a hotzone.

If two or more regions overlap, the region defined first in the map definition takes precedence over subsequent regions. This means that AREA elements with NOHREF should generally be placed before ones with the HREF attribute.

The ALT attribute is used to provide text labels which can be displayed in the status line as the mouse or other pointing device is moved over hotzones, or for constructing a textual menu for nongraphical user agents. Authors are *strongly recommended* to provide meaningful ALT attributes to support interoperability with speech-based or text-only user agents.

Sample SGML Open Catalog for HTML 3.2

This can be used with an SGML parser like nsgmls to verify that files conform to the HTML 3.2 DTD. It assumes that the DTD has been saved as the file HTML32.dtd and that the Latin-1 entities are in the file ISOlat1.ent.

```
-- html32.soc: catalog for parsing HTML 3.2 documents --
SGMLDECL "HTML32.dcl"
PUBLIC "-//W3C//DTD HTML 3.2 Final//EN" HTML32.dtd
PUBLIC "-//W3C//DTD HTML 3.2 Draft//EN" HTML32.dtd
PUBLIC "-//W3C//DTD HTML 3.2//EN" HTML32.dtd
PUBLIC "ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML" ISOlat1.ent
```

SGML Declaration for HTML 3.2

This uses the eight-bit ISO Latin-1 character set. The size limits on properties like literals and tag names have been considerably increased from their HTML 2.0 values, but it is recommended that user agents avoid imposing arbitrary length limits.

IPR2024-00143
Apple EX1011 Page 484

```
<!SGML  "ISO 8879:1986"
  --
       SGML Declaration for HyperText Markup Language version 3.2

       With support for ISO Latin-1 and increased limits
       for tag and literal lengths etc.
  --

  CHARSET
        BASESET  "ISO 646:1983//CHARSET
                  International Reference Version
                  (IRV)//ESC 2/5 4/0"
        DESCSET  0   9   UNUSED
                 9   2   9
                 11  2   UNUSED
                 13  1   13
                 14  18  UNUSED
                 32  95  32
                 127 1   UNUSED
        BASESET  "ISO Registration Number 100//CHARSET
                  ECMA-94 Right Part of
                  Latin Alphabet Nr. 1//ESC 2/13 4/1"
        DESCSET  128  32   UNUSED
                 160  96    32

  CAPACITY   SGMLREF
             TOTALCAP        200000
             GRPCAP          150000
             ENTCAP          150000

  SCOPE    DOCUMENT
  SYNTAX
     SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
            17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127
     BASESET  "ISO 646:1983//CHARSET
               International Reference Version
               (IRV)//ESC 2/5 4/0"
     DESCSET  0 128 0

     FUNCTION
             RE            13
             RS            10
             SPACE         32
             TAB SEPCHAR    9

     NAMING   LCNMSTRT ""
              UCNMSTRT ""
              LCNMCHAR ".-"
              UCNMCHAR ".-"
              NAMECASE GENERAL YES
                       ENTITY  NO
     DELIM    GENERAL  SGMLREF
              SHORTREF SGMLREF
     NAMES    SGMLREF
```



```
       QUANTITY SGMLREF
                ATTSPLEN 65536
                LITLEN   65536
                NAMELEN  65536
                PILEN    65536
                TAGLVL   100
                TAGLEN   65536
                GRPGTCNT 150
                GRPCNT   64

    FEATURES
      MINIMIZE
        DATATAG  NO
        OMITTAG  YES
        RANK     NO
        SHORTTAG YES
      LINK
        SIMPLE   NO
        IMPLICIT NO
        EXPLICIT NO
      OTHER
        CONCUR   NO
        SUBDOC   NO
        FORMAL   YES
    APPINFO      NONE
  >
```

HTML 3.2 Document Type Definition

```
<!--
        W3C Document Type Definition for the HyperText Markup Language
        version 3.2 as ratified by a vote of W3C member companies.
        For more information on W3C look at  URL http://www.w3.org/

        Date: Tuesday January 14th 1997

        Author: Dave Raggett <dsr@w3.org>

        HTML 3.2 aims to capture recommended practice as of early '96
        and as such to be used as a replacement for HTML 2.0 (RFC 1866).
        Widely deployed rendering attributes are included where they
        have been shown to be interoperable. SCRIPT and STYLE are
        included to smooth the introduction of client-side scripts
        and style sheets. Browsers must avoid showing the contents
        of these element Otherwise support for them is not required.
        ID, CLASS and STYLE attributes are not included in this version
        of HTML.
-->

<!ENTITY % HTML.Version
        "-//W3C//DTD HTML 3.2 Final//EN"

        -- Typical usage:
```


IPR2024-00143
Apple EX1011 Page 485

```
            <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
            <html>
            ...
            </html>
        --
        >

<!--================== Deprecated Features Switch ==========================-->

<!ENTITY % HTML.Deprecated "INCLUDE">

<!--================== Imported Names ======================================-->

<!ENTITY % Content-Type "CDATA"
        -- meaning a MIME content type, as per RFC1521
        -->

<!ENTITY % HTTP-Method "GET | POST"
        -- as per HTTP specification
        -->

<!ENTITY % URL "CDATA"
        -- The term URL means a CDATA attribute
           whose value is a Uniform Resource Locator,
           See RFC1808 (June 95) and RFC1738 (Dec 94).
        -->

<!-- Parameter Entities -->

<!ENTITY % head.misc "SCRIPT|STYLE|META|LINK" -- repeatable head elements -->

<!ENTITY % heading "H1|H2|H3|H4|H5|H6">

<!ENTITY % list "UL | OL |  DIR | MENU">

<![ %HTML.Deprecated [
    <!ENTITY % preformatted "PRE | XMP | LISTING">
]]>

<!ENTITY % preformatted "PRE">

<!--================ Character mnemonic entities ==========================-->

<!ENTITY % ISOlat1 PUBLIC
       "ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML">
%ISOlat1;

<!--================ Entities for special symbols ==========================-->
<!-- &trade and &cbsp are not widely deployed and so not included here -->

<!ENTITY amp    CDATA "&#38;"   -- ampersand          -->
<!ENTITY gt     CDATA "&#62;"   -- greater than       -->
<!ENTITY lt     CDATA "&#60;"   -- less than          -->
```



```
<!--==================== Text Markup ========================================-->

<!ENTITY % font "TT | I | B  | U | STRIKE | BIG | SMALL | SUB | SUP">

<!ENTITY % phrase "EM | STRONG | DFN | CODE | SAMP | KBD | VAR | CITE">

<!ENTITY % special "A | IMG | APPLET | FONT | BASEFONT | BR | SCRIPT | MAP">

<!ENTITY % form "INPUT | SELECT | TEXTAREA">

<!ENTITY % text "#PCDATA | %font | %phrase | %special | %form">

<!ELEMENT (%font|%phrase) - - (%text)*>

<!-- there are also 16 widely known color names although
  the resulting colors are implementation dependent:

   aqua, black, blue, fuchsia, gray, green, lime, maroon,
   navy, olive, purple, red, silver, teal, white, and yellow

 These colors were originally picked as being the standard
 16 colors supported with the Windows VGA palette.
 -->

<!ELEMENT FONT - - (%text)*     -- local change to font -->
<!ATTLIST FONT
    size    CDATA   #IMPLIED    -- [+]nn e.g. size="+1", size=4 --
    color   CDATA   #IMPLIED    -- #RRGGBB in hex, e.g. red: color="#FF0000" --
    >

<!ELEMENT BASEFONT - O EMPTY    -- base font size (1 to 7)-->
<!ATTLIST BASEFONT
    size    CDATA   #IMPLIED    -- e.g. size=3 --
    >

<!ELEMENT BR    - O EMPTY    -- forced line break -->
<!ATTLIST BR
        clear (left|all|right|none) none -- control of text flow --
        >

<!--================== HTML content models ================================-->
<!--
    HTML has three basic content models:

        %text       character level elements and text strings
        %flow       block-like elements e.g. paragraphs and lists
        %bodytext   as %flow plus headers H1-H6 and ADDRESS
-->

<!ENTITY % block
     "P | %list | %preformatted | DL | DIV | CENTER |
      BLOCKQUOTE | FORM | ISINDEX | HR | TABLE">

<!-- %flow is used for DD and LI -->

<!ENTITY % flow "(%text | %block)*">
```

B HTML 3.2 REFERENCE SPECIFICATION

IPR2024-00143
Apple EX1011 Page 486

```
<!--=================== Document Body ====================================-->

<!ENTITY % body.content "(%heading ¦ %text ¦ %block ¦ ADDRESS)*">

<!ENTITY % color "CDATA" -- a color specification: #HHHHHH @@ details? -->

<!ENTITY % body-color-attrs "
        bgcolor %color #IMPLIED
        text %color #IMPLIED
        link %color #IMPLIED
        vlink %color #IMPLIED
        alink %color #IMPLIED
        ">

<!ELEMENT BODY O O  %body.content>
<!ATTLIST BODY
        background %URL #IMPLIED  -- texture tile for document background --
        %body-color-attrs;  -- bgcolor, text, link, vlink, alink --
        >

<!ENTITY % address.content "((%text;) ¦ P)*">

<!ELEMENT ADDRESS - - %address.content>

<!ELEMENT DIV - - %body.content>
<!ATTLIST DIV
        align   (left¦center¦right) #IMPLIED -- alignment of following text --
        >

<!-- CENTER is a shorthand for DIV with ALIGN=CENTER -->
<!ELEMENT center - - %body.content>

<!--=================== The Anchor Element ================================-->

<!ELEMENT A - - (%text)* -(A)>
<!ATTLIST A
        name    CDATA   #IMPLIED    -- named link end --
        href    %URL    #IMPLIED    -- URL for linked resource --
        rel     CDATA   #IMPLIED    -- forward link types --
        rev     CDATA   #IMPLIED    -- reverse link types --
        title   CDATA   #IMPLIED    -- advisory title string --
        >

<!--=================== Client-side image maps ============================-->

<!-- These can be placed in the same document or grouped in a
     separate document although this isn't yet widely supported -->

<!ENTITY % SHAPE "(rect¦circle¦poly)">
<!ENTITY % COORDS "CDATA" -- comma separated list of numbers -->

<!ELEMENT MAP - - (AREA)*>
<!ATTLIST MAP
    name    CDATA   #IMPLIED
    >
```



```
<!ELEMENT AREA - O EMPTY>
<!ATTLIST AREA
    shape   %SHAPE  rect
    coords  %COORDS #IMPLIED  -- defines coordinates for shape --
    href    %URL    #IMPLIED  -- this region acts as hypertext link --
    nohref (nohref) #IMPLIED  -- this region has no action --
    alt     CDATA   #REQUIRED -- needed for non-graphical user agents --
    >

<!—=================== The LINK Element ==================================—>

<!ENTITY % Types "CDATA"
        -- See Internet Draft: draft-ietf-html-relrev-00.txt
           LINK has been part of HTML since the early days
           although few browsers as yet take advantage of it.

           Relationship values can be used in principle:

                a) for document specific toolbars/menus when used
                   with the LINK element in the document head:
                b) to link to a separate style sheet
                c) to make a link to a script
                d) by stylesheets to control how collections of
                   html nodes are rendered into printed documents
                e) to make a link to a printable version of this document
                   e.g. a postscript or pdf version
-->

<!ELEMENT LINK - O EMPTY>
<!ATTLIST LINK
        href    %URL    #IMPLIED    -- URL for linked resource --
        rel     %Types  #IMPLIED    -- forward link types --
        rev     %Types  #IMPLIED    -- reverse link types --
        title   CDATA   #IMPLIED    -- advisory title string --
        >

<!—=================== Images ============================================-->

<!ENTITY % Length "CDATA"   -- nn for pixels or nn% for percentage length -->
<!ENTITY % Pixels "NUMBER"  -- integer representing length in pixels -->

<!-- Suggested widths are used for negotiating image size
     with the module responsible for painting the image.
     align=left or right cause image to float to margin
     and for subsequent text to wrap around image -->

<!ENTITY % IAlign "(top¦middle¦bottom¦left¦right)">

<!ELEMENT IMG    - O EMPTY --  Embedded image -->
<!ATTLIST IMG
        src     %URL     #REQUIRED  -- URL of image to embed --
        alt     CDATA    #IMPLIED   -- for display in place of image --
        align   %IAlign  #IMPLIED   -- vertical or horizontal alignment --
```


IPR2024-00143
Apple EX1011 Page 487

```
        height  %Pixels  #IMPLIED   -- suggested height in pixels --
        width   %Pixels  #IMPLIED   -- suggested width in pixels --
        border  %Pixels  #IMPLIED   -- suggested link border width --
        hspace  %Pixels  #IMPLIED   -- suggested horizontal gutter --
        vspace  %Pixels  #IMPLIED   -- suggested vertical gutter --
        usemap  %URL     #IMPLIED   -- use client-side image map --
        ismap   (ismap)  #IMPLIED   -- use server image map --
        >

<!-- USEMAP points to a MAP element which may be in this document
  or an external document, although the latter is not widely supported -->

<!--=================== Java APPLET tag ==================================-->
<!--
  This tag is supported by all Java enabled browsers. Applet resources
  (including their classes) are normally loaded relative to the document
  URL (or <BASE> element if it is defined). The CODEBASE attribute is used
  to change this default behavior. If the CODEBASE attribute is defined then
  it specifies a different location to find applet resources. The value
  can be an absolute URL or a relative URL. The absolute URL is used as is
  without modification and is not effected by the documents <BASE> element.
  When the codebase attribute is relative, then it is relative to the
  document URL (or <BASE> tag if defined).
-->
<!ELEMENT APPLET - - (PARAM | %text)*>
<!ATTLIST APPLET
        codebase %URL     #IMPLIED   -- code base --
        code     CDATA    #REQUIRED  -- class file --
        alt      CDATA    #IMPLIED   -- for display in place of applet --
        name     CDATA    #IMPLIED   -- applet name --
        width    %Pixels  #REQUIRED  -- suggested width in pixels --
        height   %Pixels  #REQUIRED  -- suggested height in pixels --
        align    %IAlign  #IMPLIED   -- vertical or horizontal alignment --
        hspace   %Pixels  #IMPLIED   -- suggested horizontal gutter --
        vspace   %Pixels  #IMPLIED   -- suggested vertical gutter --
        >

<!ELEMENT PARAM - O EMPTY>
<!ATTLIST PARAM
        name    NMTOKEN   #REQUIRED  -- The name of the parameter --
        value   CDATA     #IMPLIED   -- The value of the parameter --
        >

<!--
Here is an example:

    <applet codebase="applets/NervousText"
        code=NervousText.class
        width=300
        height=50>
    <param name=text value="Java is Cool!">
    <img src=sorry.gif alt="This looks better with Java support">
    </applet>
-->
```



```
<!--=================== Horizontal Rule ==================================-->

<!ELEMENT HR    - O EMPTY>
<!ATTLIST HR
        align (left|right|center) #IMPLIED
        noshade (noshade) #IMPLIED
        size  %Pixels #IMPLIED
        width %Length #IMPLIED
        >
<!--=================== Paragraphs========================================-->

<!ELEMENT P     - O (%text)*>
<!ATTLIST P
        align  (left|center|right) #IMPLIED
        >

<!--=================== Headings ==========================================-->

<!--
  There are six levels of headers from H1 (the most important)
  to H6 (the least important).
-->

<!ELEMENT ( %heading )  - -  (%text;)*>
<!ATTLIST ( %heading )
        align  (left|center|right) #IMPLIED
        >

<!--=================== Preformatted Text ================================-->

<!-- excludes images and changes in font size -->

<!ENTITY % pre.exclusion "IMG|BIG|SMALL|SUB|SUP|FONT">

<!ELEMENT PRE - - (%text)* -(%pre.exclusion)>
<!ATTLIST PRE
        width NUMBER #implied -- is this widely supported? --
        >

<![ %HTML.Deprecated [

<!ENTITY % literal "CDATA"
        -- historical, non-conforming parsing mode where
           the only markup signal is the end tag
           in full
        -->

<!ELEMENT (XMP|LISTING) - -  %literal>
<!ELEMENT PLAINTEXT - O %literal>

]]>

<!--=================== Block-like Quotes ================================-->

<!ELEMENT BLOCKQUOTE - - %body.content>
```

B HTML 3.2 REFERENCE SPECIFICATION

IPR2024-00143
Apple EX1011 Page 488

```
<!--=================== Lists ==============================================-->

<!--
    HTML 3.2 allows you to control the sequence number for ordered lists.
    You can set the sequence number with the START and VALUE attributes.
    The TYPE attribute may be used to specify the rendering of ordered
    and unordered lists.
-->

<!-- definition lists - DT for term, DD for its definition -->

<!ELEMENT DL    - -  (DT|DD)+>
<!ATTLIST DL
        compact (compact) #IMPLIED -- more compact style --
        >

<!ELEMENT DT - O  (%text)*>
<!ELEMENT DD - O  %flow;>

<!-- Ordered lists OL, and unordered lists UL -->
<!ELEMENT (OL|UL) - -  (LI)+>

<!--
       Numbering style
    1   arabic numbers     1, 2, 3, ...
    a   lower alpha         a, b, c, ...
    A   upper alpha         A, B, C, ...
    i   lower roman         i, ii, iii, ...
    I   upper roman         I, II, III, ...

    The style is applied to the sequence number which by default
    is reset to 1 for the first list item in an ordered list.

    This can't be expressed directly in SGML due to case folding.
-->

<!ENTITY % OLStyle "CDATA" -- constrained to: [1|a|A|i|I] -->

<!ATTLIST OL -- ordered lists --
        type      %OLStyle   #IMPLIED   -- numbering style --
        start     NUMBER     #IMPLIED   -- starting sequence number --
        compact  (compact)   #IMPLIED   -- reduced interitem spacing --
        >

<!-- bullet styles -->

<!ENTITY % ULStyle "disc|square|circle">

<!ATTLIST UL -- unordered lists --
        type    (%ULStyle)   #IMPLIED   -- bullet style --
        compact (compact)    #IMPLIED   -- reduced interitem spacing --
        >

<!ELEMENT (DIR|MENU) - -  (LI)+ -(%block)>
<!ATTLIST DIR
        compact (compact) #IMPLIED
        >
```



```
<!ATTLIST MENU
        compact (compact) #IMPLIED
        >

<!-- <DIR>              Directory list                  -->
<!-- <DIR COMPACT>      Compact list style              -->
<!-- <MENU>             Menu list                       -->
<!-- <MENU COMPACT>     Compact list style              -->

<!-- The type attribute can be used to change the bullet style
     in unordered lists and the numbering style in ordered lists -->

<!ENTITY % LIStyle "CDATA" -- constrained to: "(%ULStyle|%OLStyle)" -->

<!ELEMENT LI - O %flow -- list item -->
<!ATTLIST LI
        type    %LIStyle     #IMPLIED   -- list item style --
        value    NUMBER      #IMPLIED   -- reset sequence number --
        >

<!--================ Forms ==================================================-->

<!ELEMENT FORM - - %body.content -(FORM)>
<!ATTLIST FORM
        action %URL #IMPLIED  -- server-side form handler --
        method (%HTTP-Method) GET -- see HTTP specification --
        enctype %Content-Type; "application/x-www-form-urlencoded"
        >

<!ENTITY % InputType
        "(TEXT | PASSWORD | CHECKBOX | RADIO | SUBMIT
            | RESET | FILE | HIDDEN | IMAGE)">

<!ELEMENT INPUT - O EMPTY>
<!ATTLIST INPUT
        type %InputType TEXT      -- what kind of widget is needed --
        name  CDATA   #IMPLIED   -- required for all but submit and reset --
        value CDATA   #IMPLIED   -- required for radio and checkboxes --
        checked (checked) #IMPLIED -- for radio buttons and check boxes --
        size CDATA    #IMPLIED   -- specific to each type of field --
        maxlength NUMBER #IMPLIED -- max chars allowed in text fields --
        src   %URL    #IMPLIED   -- for fields with background images --
        align %IAlign #IMPLIED   -- vertical or horizontal alignment --
        >

<!ELEMENT SELECT - - (OPTION+)>
<!ATTLIST SELECT
        name CDATA #REQUIRED
        size NUMBER #IMPLIED
        multiple (multiple) #IMPLIED
        >

<!ELEMENT OPTION - O (#PCDATA)*>
<!ATTLIST OPTION
        selected (selected) #IMPLIED
        value  CDATA  #IMPLIED -- defaults to element content --
        >
```


IPR2024-00143
Apple EX1011 Page 489

```
<!-- Multi-line text input field. -->

<!ELEMENT TEXTAREA - - (#PCDATA)*>
<!ATTLIST TEXTAREA
        name CDATA #REQUIRED
        rows NUMBER #REQUIRED
        cols NUMBER #REQUIRED
        >

<!--======================= Tables ========================================-->

<!-- Widely deployed subset of the full table standard, see RFC 1942
     e.g. at http://www.ics.uci.edu/pub/ietf/html/rfc1942.txt -->

<!-- horizontal placement of table relative to window -->
<!ENTITY % Where "(left|center|right)">

<!-- horizontal alignment attributes for cell contents -->
<!ENTITY % cell.halign
        "align  (left|center|right) #IMPLIED"
        >

<!-- vertical alignment attributes for cell contents -->
<!ENTITY % cell.valign
        "valign  (top|middle|bottom)  #IMPLIED"
        >

<!ELEMENT table - - (caption?, tr+)>
<!ELEMENT tr - O (th|td)*>
<!ELEMENT (th|td) - O %body.content>

<!ATTLIST table                         -- table element --
        align     %Where;   #IMPLIED  -- table position relative to window --
        width     %Length   #IMPLIED  -- table width relative to window --
        border    %Pixels   #IMPLIED  -- controls frame width around table --
        cellspacing %Pixels #IMPLIED  -- spacing between cells --
        cellpadding %Pixels #IMPLIED  -- spacing within cells --
        >

<!ELEMENT CAPTION - - (%text;)* -- table or figure caption -->
<!ATTLIST CAPTION
        align (top|bottom) #IMPLIED
        >

<!ATTLIST tr                            -- table row --
        %cell.halign;                   -- horizontal alignment in cells --
        %cell.valign;                   -- vertical alignment in cells --
        >

<!ATTLIST (th|td)                       -- header or data cell --
        nowrap (nowrap)  #IMPLIED  -- suppress word wrap --
        rowspan NUMBER   1         -- number of rows spanned by cell --
        colspan NUMBER   1         -- number of cols spanned by cell --
        %cell.halign;              -- horizontal alignment in cell --
        %cell.valign;              -- vertical alignment in cell --
        width   %Pixels  #IMPLIED  -- suggested width for cell --
        height  %Pixels  #IMPLIED  -- suggested height for cell --
        >
```



```
<!--================ Document Head ========================================-->

<!-- %head.misc defined earlier on as "SCRIPT|STYLE|META|LINK" -->

<!ENTITY % head.content "TITLE & ISINDEX? & BASE?">

<!ELEMENT HEAD O O  (%head.content) +(%head.misc)>

<!ELEMENT TITLE - -  (#PCDATA)* -(%head.misc)
          -- The TITLE element is not considered part of the flow of text.
             It should be displayed, for example as the page header or
             window title.
          -->

<!ELEMENT ISINDEX - O EMPTY>
<!ATTLIST ISINDEX
        prompt CDATA #IMPLIED -- prompt message -->

<!--
    The BASE element gives an absolute URL for dereferencing relative
    URLs, e.g.

         <BASE href="http://foo.com/index.html">
         ...
         <IMG SRC="images/bar.gif">

    The image is deferenced to

         http://foo.com/images/bar.gif

   In the absence of a BASE element the document URL should be used.
   Note that this is not necessarily the same as the URL used to
   request the document, as the base URL may be overridden by an HTTP
   header accompanying the document.
-->

<!ELEMENT BASE - O EMPTY>
<!ATTLIST BASE
        href %URL  #REQUIRED
        >

<!ELEMENT META - O EMPTY -- Generic Metainformation -->
<!ATTLIST META
        http-equiv  NAME    #IMPLIED  -- HTTP response header name  --
        name        NAME    #IMPLIED  -- metainformation name       --
        content     CDATA   #REQUIRED -- associated information     --
        >

<!-- SCRIPT/STYLE are place holders for transition to next version of HTML -->

<!ELEMENT STYLE  - - CDATA -- placeholder for style info -->
<!ELEMENT SCRIPT - - CDATA -- placeholder for script statements -->

<!-- ELEMENT STYLE  - - (#PCDATA)*  -(%head.misc) -- style info -->
<!-- ELEMENT SCRIPT - - (#PCDATA)*  -(%head.misc) -- script statements -->
```


IPR2024-00143
Apple EX1011 Page 490

```
<!--================ Document Structure ==================================-->

<!ENTITY % version.attr "VERSION CDATA #FIXED '%HTML.Version;'">

<![ %HTML.Deprecated [
    <!ENTITY % html.content "HEAD, BODY, PLAINTEXT?">
]]>

<!ENTITY % html.content "HEAD, BODY">

<!ELEMENT HTML O O  (%html.content)>
<!ATTLIST HTML
        %version.attr;
        >
```

Character Entities for ISO Latin-1

```
<!-- (C) International Organization for Standardization 1986
     Permission to copy in any form is granted for use with
     conforming SGML systems and applications as defined in
     ISO 8879, provided this notice is included in all copies.
     This has been extended for use with HTML to cover the full
     set of codes in the range 160-255 decimal.
-->
<!-- Character entity set. Typical invocation:
     <!ENTITY % ISOlat1 PUBLIC
       "ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML">
     %ISOlat1;
-->
    <!ENTITY nbsp   CDATA " " -- no-break space -->
    <!ENTITY iexcl  CDATA "&#161;" -- inverted exclamation mark -->
    <!ENTITY cent   CDATA "&#162;" -- cent sign -->
    <!ENTITY pound  CDATA "&#163;" -- pound sterling sign -->
    <!ENTITY curren CDATA "&#164;" -- general currency sign -->
    <!ENTITY yen    CDATA "&#165;" -- yen sign -->
    <!ENTITY brvbar CDATA "&#166;" -- broken (vertical) bar -->
    <!ENTITY sect   CDATA "&#167;" -- section sign -->
    <!ENTITY uml    CDATA "&#168;" -- umlaut (dieresis) -->
    <!ENTITY copy   CDATA "&#169;" -- copyright sign -->
    <!ENTITY ordf   CDATA "&#170;" -- ordinal indicator, feminine -->
    <!ENTITY laquo  CDATA "&#171;" -- angle quotation mark, left -->
    <!ENTITY not    CDATA "&#172;" -- not sign -->
    <!ENTITY shy    CDATA "&#173;" -- soft hyphen -->
    <!ENTITY reg    CDATA "&#174;" -- registered sign -->
    <!ENTITY macr   CDATA "&#175;" -- macron -->
    <!ENTITY deg    CDATA "&#176;" -- degree sign -->
    <!ENTITY plusmn CDATA "&#177;" -- plus-or-minus sign -->
    <!ENTITY sup2   CDATA "&#178;" -- superscript two -->
    <!ENTITY sup3   CDATA "&#179;" -- superscript three -->
    <!ENTITY acute  CDATA "&#180;" -- acute accent -->
    <!ENTITY micro  CDATA "&#181;" -- micro sign -->
    <!ENTITY para   CDATA "&#182;" -- pilcrow (paragraph sign) -->
    <!ENTITY middot CDATA "&#183;" -- middle dot -->
    <!ENTITY ccedil CDATA "&#184;" -- small c, cedilla -->
    <!ENTITY sup1   CDATA "&#185;" -- superscript one -->
    <!ENTITY ordm   CDATA "&#186;" -- ordinal indicator, masculine -->
    <!ENTITY raquo  CDATA "&#187;" -- angle quotation mark, right -->
    <!ENTITY frac14 CDATA "&#188;" -- fraction one-quarter -->
    <!ENTITY frac12 CDATA "&#189;" -- fraction one-half -->
```



```
    <!ENTITY frac34 CDATA "&#190;" -- fraction three-quarters -->
    <!ENTITY iquest CDATA "&#191;" -- inverted question mark -->
    <!ENTITY Agrave CDATA "&#192;" -- capital A, grave accent -->
    <!ENTITY Aacute CDATA "&#193;" -- capital A, acute accent -->
    <!ENTITY Acirc  CDATA "&#194;" -- capital A, circumflex accent -->
    <!ENTITY Atilde CDATA "&#195;" -- capital A, tilde -->
    <!ENTITY Auml   CDATA "&#196;" -- capital A, dieresis or umlaut mark -->
    <!ENTITY Aring  CDATA "&#197;" -- capital A, ring -->
    <!ENTITY AElig  CDATA "&#198;" -- capital AE diphthong (ligature) -->
    <!ENTITY Ccedil CDATA "&#199;" -- capital C, cedilla -->
    <!ENTITY Egrave CDATA "&#200;" -- capital E, grave accent -->
    <!ENTITY Eacute CDATA "&#201;" -- capital E, acute accent -->
    <!ENTITY Ecirc  CDATA "&#202;" -- capital E, circumflex accent -->
    <!ENTITY Euml   CDATA "&#203;" -- capital E, dieresis or umlaut mark -->
    <!ENTITY Igrave CDATA "&#204;" -- capital I, grave accent -->
    <!ENTITY Iacute CDATA "&#205;" -- capital I, acute accent -->
    <!ENTITY Icirc  CDATA "&#206;" -- capital I, circumflex accent -->
    <!ENTITY Iuml   CDATA "&#207;" -- capital I, dieresis or umlaut mark -->
    <!ENTITY ETH    CDATA "&#208;" -- capital Eth, Icelandic -->
    <!ENTITY Ntilde CDATA "&#209;" -- capital N, tilde -->
    <!ENTITY Ograve CDATA "&#210;" -- capital O, grave accent -->
    <!ENTITY Oacute CDATA "&#211;" -- capital O, acute accent -->
    <!ENTITY Ocirc  CDATA "&#212;" -- capital O, circumflex accent -->
    <!ENTITY Otilde CDATA "&#213;" -- capital O, tilde -->
    <!ENTITY Ouml   CDATA "&#214;" -- capital O, dieresis or umlaut mark -->
    <!ENTITY times  CDATA "&#215;" -- multiply sign -->
    <!ENTITY Oslash CDATA "&#216;" -- capital O, slash -->
    <!ENTITY Ugrave CDATA "&#217;" -- capital U, grave accent -->
    <!ENTITY Uacute CDATA "&#218;" -- capital U, acute accent -->
    <!ENTITY Ucirc  CDATA "&#219;" -- capital U, circumflex accent -->
    <!ENTITY Uuml   CDATA "&#220;" -- capital U, dieresis or umlaut mark -->
    <!ENTITY Yacute CDATA "&#221;" -- capital Y, acute accent -->
    <!ENTITY THORN  CDATA "&#222;" -- capital THORN, Icelandic -->
    <!ENTITY szlig  CDATA "&#223;" -- small sharp s, German (sz ligature) -->
    <!ENTITY agrave CDATA "&#224;" -- small a, grave accent -->
    <!ENTITY aacute CDATA "&#225;" -- small a, acute accent -->
    <!ENTITY acirc  CDATA "&#226;" -- small a, circumflex accent -->
    <!ENTITY atilde CDATA "&#227;" -- small a, tilde -->
    <!ENTITY auml   CDATA "&#228;" -- small a, dieresis or umlaut mark -->
    <!ENTITY aring  CDATA "&#229;" -- small a, ring -->
    <!ENTITY aelig  CDATA "&#230;" -- small ae diphthong (ligature) -->
    <!ENTITY ccedil CDATA "&#231;" -- small c, cedilla -->
    <!ENTITY egrave CDATA "&#232;" -- small e, grave accent -->
    <!ENTITY eacute CDATA "&#233;" -- small e, acute accent -->
    <!ENTITY ecirc  CDATA "&#234;" -- small e, circumflex accent -->
    <!ENTITY euml   CDATA "&#235;" -- small e, dieresis or umlaut mark -->
    <!ENTITY igrave CDATA "&#236;" -- small i, grave accent -->
    <!ENTITY iacute CDATA "&#237;" -- small i, acute accent -->
    <!ENTITY icirc  CDATA "&#238;" -- small i, circumflex accent -->
    <!ENTITY iuml   CDATA "&#239;" -- small i, dieresis or umlaut mark -->
    <!ENTITY eth    CDATA "&#240;" -- small eth, Icelandic -->
    <!ENTITY ntilde CDATA "&#241;" -- small n, tilde -->
    <!ENTITY ograve CDATA "&#242;" -- small o, grave accent -->
    <!ENTITY oacute CDATA "&#243;" -- small o, acute accent -->
    <!ENTITY ocirc  CDATA "&#244;" -- small o, circumflex accent -->
    <!ENTITY otilde CDATA "&#245;" -- small o, tilde -->
    <!ENTITY ouml   CDATA "&#246;" -- small o, dieresis or umlaut mark -->
    <!ENTITY divide CDATA "&#247;" -- divide sign -->
    <!ENTITY oslash CDATA "&#248;" -- small o, slash -->
    <!ENTITY ugrave CDATA "&#249;" -- small u, grave accent -->
```


IPR2024-00143
Apple EX1011 Page 491

```
<!ENTITY uacute CDATA "&#250;" -- small u, acute accent -->
<!ENTITY ucirc  CDATA "&#251;" -- small u, circumflex accent -->
<!ENTITY uuml   CDATA "&#252;" -- small u, dieresis or umlaut mark -->
<!ENTITY yacute CDATA "&#253;" -- small y, acute accent -->
<!ENTITY thorn  CDATA "&#254;" -- small thorn, Icelandic -->
<!ENTITY yuml   CDATA "&#255;" -- small y, dieresis or umlaut mark -->
```

Table of Printable Latin-1 Character Codes

0	32		64	@	96	‘	128		160		192	À	224	à
1	33	!	65	A	97	a	129		161	¡	193	Á	225	á
2	34	"	66	B	98	b	130		162	¢	194	Â	226	â
3	35	#	67	C	99	c	131		163	£	195	Ã	227	ã
4	36	$	68	D	100	d	132		164	¤	196	Ä	228	ä
5	37	%	69	E	101	e	133		165	¥	197	Å	229	å
6	38	&	70	F	102	f	134		166	¦	198	Æ	230	æ
7	39	’	71	G	103	g	135		167	§	199	Ç	231	ç
8	40	(	72	H	104	h	136		168	¨	200	È	232	è
9	41	)	73	I	105	i	137		169	©	201	É	233	é
10	42	*	74	J	106	j	138		170	ª	202	Ê	234	ê
11	43	+	75	K	107	k	139		171	«	203	Ë	235	ë
12	44	,	76	L	108	l	140		172	¬	204	Ì	236	ì
13	45	-	77	M	109	m	141		173	-	205	Í	237	í
14	46	.	78	N	110	n	142		174	®	206	Î	238	î
15	47	/	79	O	111	o	143		175	¯	207	Ï	239	ï
16	48	0	80	P	112	p	144		176	°	208	Ð	240	ð
17	49	1	81	Q	113	q	145		177	±	209	Ñ	241	ñ
18	50	2	82	R	114	r	146		178	²	210	Ò	242	ò
19	51	3	83	S	115	s	147		179	³	211	Ó	243	ó
20	52	4	84	T	116	t	148		180	´	212	Ô	244	ô
21	53	5	85	U	117	u	149		181	µ	213	Õ	245	õ
22	54	6	86	V	118	v	150		182	¶	214	Ö	246	ö
23	55	7	87	W	119	w	151		183	·	215	×	247	÷
24	56	8	88	X	120	x	152		184	¸	216	Ø	248	ø
25	57	9	89	Y	121	y	153		185	¹	217	Ù	249	ù
26	58	:	90	Z	122	z	154		186	º	218	Ú	250	ú
27	59	;	91	[	123	{	155		187	»	219	Û	251	û
28	60	<	92	\	124	|	156		188	¼	220	Ü	252	ü
29	61	=	93	]	125	}	157		189	½	221	Ý	253	ý
30	62	>	94	^	126	~	158		190	¾	222	Þ	254	þ
31	63	?	95	_	127		159		191	¿	223	ß	255	ÿ

Acknowledgments

The author would like to thank the members of the W3C HTML Editorial Review Board, members of the W3C staff, and the many other people who have contributed to this specification.

Further Reading

The World Wide Web Consortium

Further information on W3C activities and pointers to the status of work on HTML and HTTP etc. can be found at `http://www.w3.org/`. Further information on HTML in particular can be found at `http://www.w3.org/pub/WWW/MarkUp/`.

HTML 2.0 (RFC1866)

By Tim Berners-Lee and Dan Connolly, November 1995. Defines the Hypertext Markup Language Specification Version 2.0. Available from `ftp://ds.internic.net/rfc/rfc1866.txt`.

Form-Based File Upload in HTML (RFC1867)

By E. Nebel and L. Masinter, November 1995. Describes extensions to HTML 2.0 (RFC1866) to support file upload from HTML forms. Available from `ftp://ds.internic.net/rfc/rfc1867.txt`.

HTML Tables (RFC1942)

By Dave Raggett, May 1996. This defines the HTML table model. It is a superset of the table model defined by HTML 3.2. Available from `ftp://ds.internic.net/rfc/rfc1942.txt`, or as a W3C working draft at `http://www.w3.org/pub/WWW/TR/WD-tables`.

A Lexical Analyzer for HTML and Basic SGML

By Dan Connolly, June 1996. Describes lexical considerations for parsing HTML documents. Available from `http://www.w3.org/pub/WWW/TR/WD-html-lex`.

The Hypertext Transfer Protocol (HTTP)

Further information of HTTP can be found at `http://www.w3.org/pub/WWW/Protocols`.

A Standard Default Color Space for the Internet—sRGB

By Michael Stokes, Mathew Anderson, Srinivasan Chandrasekar, and Ricardo Motta, November 1996. Available from `http://www.w3.org/pub/WWW/Graphics/Color/sRGB.html`. This provides a precise definition for RGB that allows sRGB images to be reproduced accurately on different platforms and media under varying ambient lighting conditions.

IPR2024-00143
Apple EX1011 Page 492

APPENDIX C

CSS1 Quick Reference

by Rick Darnell

IN THIS APPENDIX

- Basic Syntax 946
- Style Attributes 947

IPR2024-00143
Apple EX1011 Page 493

This appendix provides an overview of the attributes with which you can control the appearance of your HTML documents through style sheets. The World Wide Web Consortium (W3C) set the current standard for style sheets as Cascading Style Sheets 1 (CSS1). W3C's complete recommendation for CSS is located at the W3C Web site at `http://www.w3.org/pub/WWW/TR/REC-CSS1`.

For more information on using style sheets, see Chapters 19 and 20, "Introducing Cascading Style Sheets" and "Cascading Style Sheet Usage."

Basic Syntax

All styles within a style sheet definition follow the same basic syntax. You'll notice that there are a lot of opportunities to add other attributes or members of a group:

```
SELECTOR[.class] [,SELECTOR2[.class2]] ...
{ attribute1: value1  [;
  attribute2: value2] [;
  ... ]               [;
  attributen: valuen] }
```

The `SELECTOR` is how the style is referenced within the rest of the HTML page. It uses one of the existing HTML tags, such as `<CODE>` or `<P>`, along with an optional `class` to create additional sub-styles. A `class` is a subset of a selector, allowing the same element to have a variety of styles. For example, you could color code block quotes to identify sources or speakers.

In addition to the standard HTML tags, you can use two other values for a selector: `first-line` and `first-letter`. The `first-line` value sets the style for the first line of text in a document or several passages within a document, such as paragraph or block quote. The `first-letter` value creates drop caps and other special effects on the first letter in a document or passage.

Groups of selectors and their classes are separated by commas. A member of the group receives the same style as any other member in the group. For example, if you wanted all headings to be displayed in red, you could list H1 through H6 with the attributes to set the color to red. All other tag attributes, such as size, would remain unaffected.

Another option is contextual selectors, which tell the browser what to do with a certain tag when found nested within the parent tag.

```
OUTER_SELECTOR INNER_SELECTOR {attribute:value}
```

This means that when the `INNER_SELECTOR` is used within the `OUTER_SELECTOR`, the style is used. Otherwise, other occurrences of `INNER_SELECTOR` are handled according to browser default.

After making all of the selector and group definitions, use a curly bracket along with a series of attributes and their values. Mate each attribute with its value with a colon and separate each pair from the next pair by a semicolon. The values within a definition, such as the name of a

typeface or a color value, are not case-sensitive. For example, for `font-family`, you can have `Garamond`, `garamond`, or `GARAMOND`, and it will all work out the same in the browser.

As with all good syntax, you can place style definitions in three ways within a document: with an embedded style sheet, with a linked style sheet, and with an inline style sheet.

Embedded Style Sheet

The `<STYLE>` tags contain an embedded style sheet. As a matter of structure, the format of an HTML page with an embedded style sheet is as follows:

```
<HTML>
<HEAD>...</HEAD>
<STYLE>...</STYLE>
<BODY>...</BODY>
</HTML>
```

The `<STYLE>` tags contain the list of selectors and styles.

Linked Style Sheet

The linked style sheet is a `.css` file that contains nothing but a set of `<STYLE>` tags and their contents. Identify the style file within an HTML document using the `<LINK>` tag in the head:

```
<HEAD>
<LINK rel=stylesheet href="filename.css" type="text/css">
</HEAD>
```

At runtime, the browser will load the style in the `.css` file and use it to format the document. If the HTML page also includes an embedded style sheet that conflicts with the linked style sheet, the embedded version also takes precedence.

Inline Style Sheet

The last option, inline style sheets, uses style sheet syntax, although it's technically not a style sheet implementation. This option uses the style sheet nomenclature to customize single incidents of tags within the document:

```
<TAG style="attribute1:value1; ...">
```

Essentially, this is a way to customize HTML tags on a case-by-case basis. When you use all three forms of syntax, they occur in a cascading form of precedence. The highest priority is inline, followed by embedded, then linked.

Style Attributes

Several classes of attributes are used within the definition for a selector. The following sections cover each of the attributes within a class.

IPR2024-00143
Apple EX1011 Page 494

Fonts

There are no current standards for typefaces and their use on different user machines, so you'll need to choose carefully and include several options to achieve the desired effect for the user.

The font-family Attribute

The font-family attribute lists font families in order of preference, separated by commas. Two types of variables are used: family name and generic family.

```
BODY {font-family: Garamond, Palatino, Serif}
```

A family name is the name of a specific typeface such as Helvetica, Garamond, Palatino, or Optima. Enclose font names with spaces in quotes, such as "Gil Sans". The generic family is one of five choices that classifies the typeface by its style and is recommended as the last option in a font-family list:

- Serif: Fonts with accents at the tips of the lines (for example, Times)
- Sans-serif: Fonts without finishing accents (for example, Helvetica)
- Cursive: Scripts that more closely resemble hand-drawn calligraphy (for example, Zapf Chancery)
- Fancy: Special-use decorative fonts (for example, Comic Book Sans)
- Monospace: Fonts that maintain uniform spacing despite letter width (for example, Courier)

The font-style Attribute

This attribute specifies the type of treatment a font receives and is represented by the values normal, italic, or oblique. The normal value is also referred to as Roman in some typeface references. The oblique value is similar to italic except that it is usually slanted manually by the system rather than by a separate style of the font, like italic.

```
BODY {font-style: italic}
```

The font-variant Attribute

Similar to font-style, this attribute sets small caps. Its two values are normal and small-caps.

```
BODY {font-variant: small-caps}
```

If there is no true small caps version of the typeface, the system will attempt to scale the capital letters to a smaller size for lowercase letters. As a last resort, the text will appear in all capitals.

The font-weight Attribute

A number of values for this attribute set the darkness or lightness of a typeface. The primary values are normal and bold. You can substitute these values with one of a list of values from 100 to 900. If a typeface includes a "medium" weight, it will correspond to 500. Bold is represented by 700.

```
BODY {font-weight: bold}
```

Two additional values are bolder and lighter, which increase the weight from the current parent weight by one level, such as 200 to 300 for bolder or 700 to 600 for lighter.

The font-size Attribute

Four methods can define the size of a font in a style—absolute size, relative size, length, or percentage.

- Absolute size: This method is represented in several ways. The first is with a value that represents its size relative to other sizes within the family (xx-small, x-small, small, medium, large, x-large, xx-large). You can also use a numerical value, such as 12pt (12 points).

  ```
  BODY {font-size: 18pt}
  ```

- Relative size: This method sets the size relative to the parent style. It can be one of two values, smaller or larger, and it adjusts the size up or down the scale of sizes. If a font doesn't include a mapping to size names, a scaling of 1.5 is recommended between sizes. For example, a 10pt font would be scaled larger to 15pt or smaller to 7pt.

  ```
  P {font-size: smaller}
  ```

- Length: This method is another form of relative size that sets the size by the scale factor of the width of an em, such as 1.5em.

  ```
  P {font-size: 2em}
  ```

- Percentage: This method is also a relative specification that multiplies the size of the parent font by the percentage value to achieve the new size, such as 150%.

  ```
  H3 {font-size: 300%}
  ```

The font Attribute

This attribute provides a shorthand for setting all of the previous attributes under one umbrella. The order of the attributes should be font-style, font-variant, font-weight, font-size, line-height, font-family. Place no commas between each of the attribute values, except for listed font families:

```
BODY {font: small-caps bold 14pt garamond, palatino, serif}
```

Color and Background

These elements set the color values for the text (foreground) and the area behind the text (background). In addition to setting a background color, you can also define a background image. All color values are defined using the same methods as the color attribute.

IPR2024-00143
Apple EX1011 Page 495

The color Attribute

This attribute defines the color of the text element and is specified using one of the color keywords (such as red). You can also define the color using a hexadecimal triplet, denoting the mix of red, green, and blue (such as rgb(255,0,0)).

```
BLOCKQUOTE {color: rgb(0,255,255)}
```

The background-color Attribute

This attribute sets the background color for a style. You can set this attribute independently of a background color for the document to enable you to highlight text in a different manner.

```
BLOCKQUOTE {background-color: blue}
```

The background-image Attribute

This attribute specifies a background image for a style element. Use it in conjunction with background-color to ensure a substitute effect if the image becomes unavailable. If the image is available, it will display on top of the background color.

```
BLOCKQUOTE {background-image: url(logo.gif)}
```

The background-repeat Attribute

If the background image should be repeated (tiled), use this attribute to define how. Its values include repeat, repeat-x, and repeat-y. The repeat value indicates that the image should be tiled normally. The repeat-x value repeats the image in a single horizontal line, and the repeat-y value repeats the image in a vertical line.

```
BLOCKQUOTE {background-image: url(logo.gif);
            background-repeat: repeat-x}
```

The background-attachment Attribute

This attribute, an extended feature of background images not seen in HTML before, sets whether the background image is attached to the foreground text (scroll) or anchored to the background (fixed). This feature is apparent only when the user scrolls across a selection of text.

```
BLOCKQUOTE {background-image: url(logo.gif);
            background-attachment: repeat-x}
```

The background-position Attribute

When you use a background image through normal HTML, the starting point is always the top left of the screen. With a style sheet, you can specify a starting point anywhere within the box that contains the style content.

You can specify the image's starting position in three ways. The first way is with key word locations. For horizontal placement, your choices are left, center, or right. For vertical placement, your choices are top, center, or bottom. Alternatively, you can represent the position as

a percentage of the available area, with 0% 0% being the top left (default) and 100% 100% being the bottom right. The last option is to specify an actual measurement in centimeters or inches.

If only one value for the placement is given, it's used as the horizontal position. If both values are given, the first is evaluated as horizontal and the second as vertical.

```
BLOCKQUOTE {background-image: url(logo.gif);
            Background-repeat: repeat-y;
            background-position: right top; }
```

The background Attribute

This shorthand attribute, similar to font, enables you to define a set of values for the background in one stop. The order is background-color, background-image, background-repeat, background-attachment, and background-position.

```
P { background: black url(logo.gif) repeat-y fixed right top }
```

Text

This set of style attributes covers the values that can affect the appearance of text, but not by directly changing the typeface. This includes values for spacing, underlining, blinking, and strike-through. It also supports some of the positioning attributes, including left and right justification and indents.

The word-spacing Attribute

This attribute indicates an addition to the default amount of space between individual words and is specified in *ems*. An *em* is the space occupied by the letter "m" and is the baseline for determining widths within a font. To return the value to its default, use 0em or normal.

```
BODY { word-spacing: 1em }
```

The letter-spacing Attribute

The letter-spacing attribute is similar to word-spacing, except that letter-spacing adds an extra bit of spacing between individual letters. In addition to the default method the browser uses to determine spacing, additional letter spacing is also affected by text alignment.

```
BODY { letter-spacing: 0.2em }
```

The text-decoration Attribute

This attribute is more closely related to its cousins in the font family. It specifies extra text flourishes, such as underline, strike-through, and blinking. The four values are none, underline, overline, line-through, and blink.

```
STR.blink { text-decoration: underline blink }
```


IPR2024-00143
Apple EX1011 Page 496

The vertical-align Attribute

This attribute sets the vertical position of the text either to an absolute reference or in relation to the parent element. It supports a range of values and keywords:

- `Baseline`: Aligns the baseline of the style with the baseline of the parent element
- `Sub`: Assigns the style to a subscript relative to the parent element
- `Super`: Assigns the style to a superscript relative to the parent element
- `Text-top`: Aligns the top of the text with the top of the parent's text
- `Text-bottom`: Aligns the bottom of the text with the bottom of the parent's text
- `Middle`: Aligns the vertical halfway point of the element with the baseline of the parent plus half of the x-height of the parent (x-height is the height of the lowercase `x` of the font)
- `Top`: Aligns the top of the element with the tallest element on the current line
- `Bottom`: Aligns the bottom of the element with the lowest element on the current line
- `(Percentage)`: Using a positive or negative percentage value, raises or lowers the element beyond the baseline of the parent

```
SUB { vertical-align: -10% }
```

The text-transform Attribute

This attribute sets the capitalization of the affected text to one of four choices: `capitalize` (first letter of every word), `uppercase` (all letters in capitals), `lowercase` (all letters in lowercase), and `none`.

```
STR.caps { text-transform: uppercase }
```

The text-align Attribute

This attribute moves beyond the standard HTML `left-right-center` alignment to provide full justification (`justify` left and right). If a browser doesn't support `justify`, it will typically substitute `left`.

```
BLOCKQUOTE { text-align: justify }
```

The text-indent Attribute

The `text-indent` attribute, specified in an absolute value measured in ems or inches, defines the amount of space that is added before the first line.

```
P { text-indent: 5em }
```


The line-height Attribute

This attribute sets the distance between adjacent baselines using a length (in ems), multiplication factor, or percentage. Factors are indicated without any units, such as `1.5`. When you use this method, the child inherits the factor, not the resulting value.

```
DIV { line-height: 1.5; font-size: 12pt }
```

In this instance, the line height becomes 18 points and the font size remains at 12 points.

Margins, Padding, and Borders

Each element created in a style sheet is presented in its own "box." All of the styles from the element inside the box are applied, although the box itself can have its own properties that define how it relates to adjoining elements on the page. Length is specified in inches (`in`), centimeters (`cm`), ems (`em`), points (`pt`), or pixels (`px`).

Box properties are divided into three basic categories. Margin properties set the border around the outside of the box, padding properties determine how much space to insert between the border and the content, and border properties define graphical lines around an element.

Additional properties of the box include its width, height, and physical position.

The margin-top, margin-bottom, margin-right, and margin-left Attributes

These four attributes set the amount of space between the element and adjoining elements, whether defined by length or percentage of parent text width or handled automatically.

```
BLOCKQUOTE { margin-top: 4em;
             Margin-bottom: auto }
```

The margin Attribute

The `margin` attribute provides a shorthand method for setting the four margin values.

When you specify the four values, they are applied, in order, to the top, right, bottom, and left. If you provide only one value, it applies to all sides. If you use two or three values, the missing values are copied from the opposite sides.

```
BLOCKQUOTE {margin: 4em 2em}
```

The padding-top, padding-bottom, padding-right, and padding-left Attributes

These attributes set the distance between the boundaries of the box and the elements inside the box. It can use any of the physical measurements or a percentage of the parent's width.

```
BLOCKQUOTE {padding-top: 110%; padding-bottom: 115%}
```


IPR2024-00143
Apple EX1011 Page 497

The padding Attribute

The padding attribute provides a shorthand method for setting the four padding values.

When you specify the four values, they are applied, in order, to the top, right, bottom, and left. If you provide only one value, it applies to all sides. If you use two or three values, the missing values are copied from the opposite sides.

```
BLOCKQUOTE {padding: 10pt 12pt}
```

The border-top, border-bottom, border-right, and border-left Attributes

These four attributes set the style and color of each border around an element. Specify styles with one of the border style keywords: none, dotted, dashed, solid, double, groove, ridge, inset, and outset. For more information on these, see the information on border-style later in this chapter.

Specify colors using a color keyword. For more information, see the border-color later in this chapter.

```
BLOCKQUOTE {border-left: solid red}
```

The border-top-width, border-bottom-width, border-right-width, and border-left-width Attributes

These attributes define a physical border around the box, similar to the border used for HTML tables. In addition to defining a specific width in ems, you can also use the keywords thin, medium, and thick. Using a measurement in ems results in a border whose width changes in relation to the size of the current font.

```
STR {border-right-width: 2pt;
     border-left-width: 2pt }
```

The border-width Attribute

The border-width attribute provides a shorthand method for setting the width of the four borders.

When you specify the four values, they are applied, in order, to the top, right, bottom, and left. If you provide only one value, it applies to all sides. If you use two or three values, the missing values are copied from the opposite sides.

```
BLOCKQUOTE {border-width: medium 0pt 0pt thick}
```

The border-color Attribute

This attribute sets the color of all four borders and uses one color keyword as its value. You cannot set the color of each side independently.

```
BLOCKQUOTE {border-color: yellow}
```


The border-style Attribute

The border's appearance can take on several different settings, represented by none, dotted, dashed, solid, double, groove, ridge, inset, and outset. The last four values are represented in 3D, if the browser supports it. Alternatively, the browser also can present all of the variations as a solid line, except none.

Like border-color, the style is applied uniformly to all four sides.

```
BLOCKQUOTE {border-style: groove}
```

The border Attribute

The border attribute provides a shorthand method for setting all of the border variables, including width, style, and color. It sets the values for all four sides at the same time, overriding any individual settings that may have been set previously for the same element.

```
BLOCKQUOTE {border: 1.5pt double black}
```

The height Attribute

This attribute sets the overall height of the bounding box that contains either the text or image element. If the content is text, scrollbars are added as needed so that all of the material is still available to the user. If the content is an image, it's scaled to fit inside the area. You can set a physical value or use auto to let the browser allocate space as needed.

```
BLOCKQUOTE {height: 100px}
```

The width Attribute

Similar to height, the width attribute sets the overall width of the bounding box that contains the element. If the content is text, scrollbars are added as needed so that all of the material is still available for the user. If both elements are used with an image and the value of one element is auto, the aspect ratio for the image is maintained.

```
BLOCKQUOTE {width: auto}
```

The float Attribute

This attribute sets a value similar to the align attribute used in HTML. The three possible values are left, right, and none. The none value allows the element to fall where it may, and the other two values force the element to the left or right of the screen with text wrapping around the opposite side.

```
BLOCKQUOTE {float: right}
```

The clear Attribute

This attribute mimics the clear attribute used with the HTML <BR> tag and uses the same keywords as float. If you use it with right or left, elements will move below any floating

IPR2024-00143
Apple EX1011 Page 498

element on that respective side. If you set it to none, floating elements are allowed on both sides.

```
BLOCKQUOTE {clear: left right}
```

Classification

These attributes control the general behavior of other elements more than actually specifying an appearance. In addition, classification includes the attributes for list items, identified in HTML with the <LI> tag.

The display Attribute

This attribute identifies when and if a style element should be used. Four keywords determine its behavior:

- Inline: A new box is created within the same line as adjoining text items and is formatted according to the size and amount of content within its borders, such as an image (IMG) or text (STR).
- Block: A new box is created relative to the surrounding elements. This is common with elements such as H1 and P.
- List-item: Similar to block, only list item markers, which behave more like inline content, are added.
- None: Turns off the display of the element in any situation, including for children of the element.

```
IMG {display: inline}
BLOCKQUOTE {display: block}
```

The white-space Attribute

The name of this attribute is a bit misleading because it relates to how spaces and line breaks are handled. The choices are normal (in which extra spaces are ignored), pre (as in preformatted HTML text), and nowrap (in which lines are broken only with <BR>).

```
BLOCKQUOTE {white-space: pre}
```

The list-style-type Attribute

This element sets the type of markers used for a list. Your choices are disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, and none. For more information on how each of these is represented onscreen, see Chapter 9, "Using Lists to Organize Information."

```
LI.outline1 {list-style-type: upper-roman}
LI.outline2 {list-style-type: upper-alpha}
LI.outline3 {list-style-type: decimal}
```


The list-style-image Attribute

In lieu of a text marker for the list item, you can also specify the URL of an image to use. If the image is unavailable, the text marker is used as default.

```
LI.general {list-style-image: url(bullet.jpg)}
```

The list-style-position Attribute

The two values for this attribute, inside and outside, determine the formatting of text following the list item marker. The outside value, the default value, lines up the additional lines of text beyond the first line with the first character in the first line. If you use the inside value, the second and following lines are justified with the list item marker.

```
LI {list-style-position: inside}
```

The list-style Attribute

This attribute is a shorthand element for the list-style-type, list-style-image, and list-style-position attributes.

```
OL {list-style: lower-alpha outside}
UL {list-style: square url(bullet.jpg) inside}
```


IPR2024-00143
Apple EX1011 Page 499

APPENDIX D

HTML Resources

by David Mayhew

IN THIS APPENDIX

- Information on HTML 960
- Browsers 962
- HTML Editors and Web Authoring Tools 964
- Usenet Newsgroups 966

IPR2024-00143
Apple EX1011 Page 500

Because the Web is constantly changing, no book can be 100 percent up-to-date, even on the day it arrives in the bookstores. Thankfully, hundreds of resources available on the Web itself are frequently revised to keep their information current.

In this appendix, you'll find a summary of many of those resources, including their URLs and a brief description. In addition to Web pages about changes in HTML, the latest browser developments, and HTML development tools, at the end of this appendix is a guide to relevant newsgroups where hundreds of messages are posted each day.

Information on HTML

This section gives you the details on several Web sites where you can find current HTML information.

Netscape DevEdge OnLine

`http://developer.netscape.com/index.html`

If you use Netscape technology to develop your Internet site or intranet, you need to visit the DevEdge site frequently. DevEdge's mission is to provide "communications, tools, support, and marketing assistance to speed the planning, development, and deployment of Internet and Intranet solutions." The site is a combination of free and fee-based content. The free access is primarily through a twice-weekly news e-mail with the latest developments in Netscape technology, news of new betas, bug fixes, and so on. Premium services include early access to betas, a quarterly CD full of Netscape tools, a Netscape ONE directory link, and peer newsgroups.

Microsoft Site Builder Network

`http://www.microsoft.com/sitebuilder/`

Here you will find a very deep resource for developing Web sites with Microsoft technology. The Site Builder Network is a multilevel program that delivers technical information, products, technologies, services, and support for the latest Internet technologies, such as new HTML extensions, ActiveX controls, and Java applets. Information is scaled to match your level of membership, but even the content available free of charge makes this site worthy of frequent visits. Of particular interest is the Workshop area, which contains hands-on information on authoring, design, programming, and server technologies.

The World Wide Web Consortium (W3C)

`http://www.w3.org/pub/WWW/`

The W3C was founded in 1994 to develop common protocols for the evolution of the World Wide Web. The Consortium is led by Tim Berners-Lee, Director of the W3C and creator of the World Wide Web; and Jean-François Abramatic, Chairman of the W3C; and is funded by commercial members. The vendor-neutral W3C works with the global community to produce specifications and reference software that is made freely available throughout the world.

Cougar

`http://www.w3.org/pub/WWW/MarkUp/Cougar/`

Cougar is W3C's next version of HTML.

HTML 3.2 Reference Specification

`http://www.w3.org/pub/WWW/TR/REC-html32.html`

This site offers the final specification for HTML 3.2, written by Dave Raggett.

HTML 2.0 Specification

`http://www.w3.org/pub/WWW/MarkUp/html-spec/html-spec_toc.html`

Tim Berners-Lee and Dan Connolly wrote the HTML 2.0 specification, located at this site.

Cascading Style Sheets, Level 1, W3C Recommendation

`http://www.w3.org/pub/WWW/TR/REC-CSS1`

This document specifies level 1 of the Cascading Style Sheet mechanism (CSS1).

Document Object Model (Dynamic HTML)

`http://www.w3.org/pub/WWW/MarkUp/DOM/`

This site offers an overview of materials related to the Document Object Model at W3C and around the Web. As mentioned earlier in this book, the Document Object Model (DOM) is a platform-independent and language-neutral interface that allows programs and scripts to dynamically access and update the structure, content, and style of Web pages.

URL Descriptions

`http://www.w3.org/hypertext/WWW/Addressing/Addressing.html`

This site presents definitions of the various types of URLs as well as discussion of URIs and URNs.

Yahoo!

`http://www.yahoo.com/Computers/World_Wide_Web`

This site exhibits links to more than 3,000 pages regarding HTML. Find links here to sites about Internet technologies such as HTML, CGI, Java, ActiveX, VBScript, VRML, and many others, as well as resources for design, programming, browser information, and server details, to name just a few.

IPR2024-00143
Apple EX1011 Page 501

HTML Writers Guild

http://hwg.org

The HTML Writers Guild is an international organization of World Wide Web page authors and Internet publishing professionals. The more than 50,000 members enjoy a wealth of HTML resources, plus access to their newsgroups. Basic membership is free and includes access to more information than many sites give you after charging a fee. Several paid membership levels are available as well, and these offer benefits such as mentoring programs, educational classes, access to the job board, and software discounts.

The Web Design Group

http://www.htmlhelp.com

This is a highly useful site for beginners and experienced developers alike. Here you will find the widely acclaimed HTML 3.2 Reference, a Cascading Style Sheets Guide, and a Character Set Overview.

The HTML Guru

http://members.aol.com/htmlguru/index.html

This site offers a collection of HTML tips, tricks, and hacks covering both document authoring and Web server management, presented in a question and answer format.

Browsers

The HTML that you create doesn't always look the way that you intended on every user's screen. The different ways that browsers interpret your HTML can be frustrating, causing your Web pages to appear flat, boring, or unreadable instead of dynamic and exciting. To counter this, you should visit the following sites frequently to keep up on the latest developments in browser technology.

Browserwatch

http://browserwatch.iworld.com

This browser offers breaking news in the browser and plug-ins industry, plus browser usage statistics and a rich library of plug-ins and ActiveX components.

Netscape

http://home.netscape.com

Here you will find Netscape Navigator, as well as a very rich source of information on Web-related issues.

Microsoft Internet Explorer

http://www.microsoft.com/ie

Get the latest version of Microsoft's Web browser here. This site also maintains an archive of press releases and links to articles relating to Internet Explorer.

Amaya

http://www.w3.org/pub/WWW/Amaya/

The follow-up to Arena, this browser was developed by Irène Vatton, Vincent Quint, and Daniel Veillard with the purpose of being a testbed for experimenting and demonstrating new specifications and extensions of Web protocols and standards.

Arena

http://www.yggdrasil.com/Products/Arena/

Arena is a graphical Web browser with origins that predate proprietary packages such as Netscape Navigator, Microsoft Internet Explorer, and Mosaic. It is the source of a number of innovations that have since been copied by other Web browsers, such as HTML tables and style sheets.

Cyberdog

http://www.cyberdog.apple.com/

This site presents a versatile tool that includes a Web browser, using Apple's OpenDoc technology.

Galahad

http://www.mcs.com/~jvwater/main.html

Galahad is an offline Web browser.

Lynx

http://www.nyu.edu/pages/wsn/subir/lynx/platforms.html

Lynx is a fully featured World Wide Web browser for users on both UNIX and VMS platforms who are connected to those systems via cursor-addressable, character-cell terminals or emulators. This text-based browser is widely used in universities, libraries, and many other situations in which there is a desire to bring the information of the World Wide Web to as wide an audience as possible.

IPR2024-00143
Apple EX1011 Page 502

NCSA Mosaic

http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html

This Web browser was developed at the National Center for Supercomputing Applications at the University of Illinois in Urbana-Champaign.

Opera

http://traviata.nta.no/opera.htm

You can customize this highly configurable browser to fit the user's needs. As a bonus, you can easily configure this browser for users with a handicap, and it is also ideal for slow machines.

SlipKnot

http://plaza.interport.net/slipknot/slipknot.html

This Web browser works without a PPP or SLIP connection.

HTML Editors and Web Authoring Tools

Many great tools are available to make writing HTML easier and more efficient. Several *tag editors* look a lot like the text editor you might be using now, but have useful features that your text editor lacks. However, the trend recently has been toward tools that display your page in a WYSIWYG window while you work. These tools make it easy to get immediate feedback on your site's appearance and allow for manipulation of the HTML at the object level.

HTML Tag Editors

First, take a look at the tag editors. The following headings list several of the more popular tools available.

BBEdit

http://www.barebones.com/bbedit.html

BBEdit 4.0.4 by Bare Bones Software is available at this site.

HomeSite

http://www.dexnet.com/homesite.html

This site is home to HomeSite 2.5, which Nick Bradbury wrote and now Allaire owns. In addition to creating HTML pages and Web sites, HomeSite also can create Cold Fusion Web applications.

HotDog

http://www.sausage.com

The HotDog Web editor by Sausage Software is located at this site.

HoTMetaL Pro

http://www.sq.com/products/hotmetal/hmp-org.htm

HoTMetaL by SoftQuad is located at this site.

Webber

http://www.csdcorp.com/webber.htm

This editor from Cerebral Systems Development includes a validator program to look for invalid HTML. Also check out Webber/Active at the same URL. This product does everything Webber does, and it also includes support for ActiveX controls.

WYSIWYG Editors

WYSIWYG editors have received more attention and more respect as powerful utilities such as FrontPage, Backstage, and PageMill have appeared. Under the following headings, you find links to a few of the most popular and most powerful products in this category.

Netscape Composer

http://home.netscape.com/try/comprod/mirror/client_download.html

Netscape Composer is the WYSIWYG editor in the Netscape Communicator suite.

Microsoft FrontPad

http://www.microsoft.com/ie/ie40/

FrontPad is the WYSIWYG Web page editor in Microsoft Internet Explorer 4. If you're not sure you want to take the plunge with FrontPage, FrontPad is a great way to find out for free. Although nobody at Microsoft would say that FrontPad is "FrontPage Lite," FrontPad does appear to be a scaled down version of the best-selling commercial product.

Microsoft FrontPage 97

http://www.microsoft.com/frontpage

FrontPage 97 provides a total Web site creation solution, including site structuring, HTML editing, site maintenance, and the FrontPage Web Server.

IPR2024-00143
Apple EX1011 Page 503

PageMill

`http://www.adobe.com/prodindex/pagemill/overview.html`

PageMill is a powerful WYSIWYG editor from Adobe, the company behind Portable Document Format (PDF) and PageMaker.

Backstage Internet Studio

`http://www.macromedia.com/software/backstage/`

Backstage is Macromedia's WYSIWYG Web development tool. In addition to easily integrating multimedia components to your Web pages, Backstage offers powerful database integration features.

Usenet Newsgroups

Usenet provides a platform for any person with Net access to express to the whole world his or her views on a topic. As you might expect, this results in a great deal of chaos. But mixed into the chaos are a lot of useful and timely ideas. The following newsgroups promise to be full of lively discussions of the latest developments in HTML, browsers, and other Web-related issues. A word of caution: Before posting your question to a newsgroup, read the group to make sure your question has not been asked and answered already. If you don't, you'll learn the unfriendly definitions of terms like *newbie* and *flame-bait*.

The `comp.infosystems.www.authoring.html` Newsgroup

At this very useful, vibrant newsgroup, you will find information on all aspects of HTML authoring.

The `alt.html` Newsgroup

The `alt.html` newsgroup is a much less structured, more relaxed HTML newsgroup.

The `comp.infosystems.www.authoring.misc` Newsgroup

This newsgroup is a forum for discussion of questions not covered in the more specific authoring newsgroups.

The `comp.infosystems.www.browsers.ms-windows` Newsgroup

This newsgroup hosts discussion of Microsoft Windows browsers.

The `comp.infosystems.www.browsers.mac` Newsgroup

This newsgroup hosts discussion of Macintosh browsers.

The `comp.infosystems.www.browsers.misc` Newsgroup

This newsgroup hosts discussion of all other browsers.

APPENDIX E

What's on the CD-ROM

IN THIS APPENDIX

- Windows Software 968
- Macintosh Software 969
- About the Software 969

IPR2024-00143
Apple EX1011 Page 504

On the *HTML Unleashed* CD-ROM, you will find the sample files that were presented in this book, along with a wealth of other applications and utilities.

NOTE

Please refer to the `readme.wri` file on the CD-ROM (Windows) or the Guide to the CD-ROM (Macintosh) for the latest listing of software.

Windows Software

ActiveX

- Microsoft ActiveX Control Pad and HTML Layout Control

Explorer

- Microsoft Internet Explorer 3

Graphics, Video, and Sound Applications

- Cell Assembler
- Goldwave sound editor, player, and recorder
- MapThis image map utility
- Paint Shop Pro
- SnagIt
- ThumbsPlus
- Web graphics from The Rocket Shop

HTML Tools

- NetObjects Fusion demo
- W3e HTML Editor
- Hot Dog 32-bit HTML editor
- HTMLed HTML editor

Utilities

- Adobe Acrobat viewer
- WinZip for Windows NT/95
- WinZip Self-Extractor

Macintosh Software

Graphics, Video, and Sound Applications

- Graphic Converter
- GIFConverter
- Fast Player
- Sparkle
- SoundApp
- Web graphics from The Rocket Shop

HTML Tools

- NetObjects Fusion demo
- BBEdit LITE
- HTML Web Weaver
- WebMap
- HTML.edit
- HTML Editor for the Macintosh

Utilities

- ZipIt for Macintosh
- ScrapIt Pro
- Adobe Acrobat

About the Software

Please read all documentation associated with a third-party product (usually contained with files named `readme.txt` or `license.txt`), and follow all guidelines.

IPR2024-00143
Apple EX1011 Page 505

I

INDEX

IPR2024-00143
Apple EX1011 Page 506

Symbols

& (ampersands), 57, 60
name/value pairs, 518
URL encoding, 520
*** (asterisks), 60**
, (commas), 60
selectors, 946
<!--> (comment tags), 40, 862
{} (curly brackets), style sheets, 946
= (equals signs), 74
URL encoding, 520
 # (number signs), URL encoding, 521
% (percent signs), 56
URL encoding, 519-521
+ (plus signs), 60
? (question marks), 60
URL encoding, 520
; (semicolons), style sheets, 946
/ (slashes)
trailing, 104
URL encoding, 520
XML, 785
| (vertical bars), 60
3-D, 630-631
plug-ins, 663-665
ray tracers, 727
POV-Ray, 727
Ray Dream Designer, 727
Sun's API, 634
VRML, 631-632
billiard table listing, 638-639
embedding, 644-645
frames, 645-646
future developments, 647-649
history, 633-634
nodes, 635
reference documents, 637
scenes, 635
version 1, 632
version 2, 632
viewers, 632
VRML Consortium, 634
Web site appearance, 632-633
7-bit ASCII, 795-796
control characters, 795
restrictions, 795
8-bit encodings, 796-797
control characters, 796
ISO 8859 standards, 796
16-bit encodings, 797-798
backward compatibility, 797
code switching, 797
32-bit encoding, 798-799
BMP (Basic Multilingual Plane), 799

A

<A> tag, 84, 223, 865-866
attributes
href, 223-224, 227
name, 228
target, 226, 343-344
title, 227-228
internal anchors, 229-232
mailto: protocol, 226-227
W3C HTML 3.2 specification, 920
above attribute, <LAYER> and <ILAYER> tags, 350
absolute text size, 949
specifying, 175
absolute widths (tables), 456
Accent Software Web site, 670
Accept header, 532
accept-charset attribute, <FORM> tag, 803-804
Accept-Charset header, 532
Accept-Encoding header, 532
Accept-Language header, 532
access
file (Java), 570
Internet
dialup, 843
gophers, 845
XML DTDs, 786-787
Access 97, 682-683
ASP, 683
dynamic approach, 682
IDC/HTX, 682-683
links, 682
static approach, 682
Web site, 682
wizards, 683
accesskey attribute, <LABEL> tag, 332
Acrobat Reader, 668
Action property (forms), 602
active links, ALINK attribute, 151
active server pages, see ASP
ActiveX, 605, 840
Control Pad, 607-608
Script Wizard, 610-613
controls, 260-262, 593, 606
align attribute, 261
Backstage Designer, 747
built-in, 606-607
class IDs, 261
codebase attribute, 261
codetype attribute, 261
data control objects, 426
including, 261
making available to users, 614
Microsoft HTML Layout Control, 615-616
multimedia, 249
navigation, 477
properties, 608-610
scripts, 610-613
third party, 607
importance of, 592
security, 614
add-ons, see plug-ins

<ADDRESS> tag, 82, 165, 869
ordered lists, 184
W3C HTML 3.2 specification, 901
addresses (IP), 848
Administrator (Backstage), 750
Adobe Acrobat Reader, 668
Adobe Illustrator, 721
features, 721
system requirements, 721
Web site, 721
Adobe Photoshop, 713-715
features, 713-714
filters, 714
resource sites, 714-715
Andy's Photoshop tips, 715
mailing list, 715
online reference guide, 715
PC Resource for Photoshop, 714
Photoshop Paradise, 715
Ultimate Compendium of Photoshop Sites, 714
Usenet newsgroup, 715
system requirements, 715
weaknesses, 714
Web site, 713
Adobe Web site, 693
agents, 840, 854
Alchemy Mindworks GIF Construction Set Web site, 256
Alert method, 601
alerts, displaying, 601
align attribute
ActiveX controls, 261
<DIV> tag, 163
headings, 156
<HR> tag, 177
<IFRAME> tag, 330
<IMG> tag, 235-238
bottom value, 237
left value, 237
middle value, 236
right value, 238
top value, 235
<MARQUEE> tag, 258
<OBJECT> tag, 334
<P> tag, 158
<TD> and <TH> tags, 212
<TR> tag, 205
align property, JavaScript style sheets, 408
aligning
background images, 950
repeating, 950
cells, 212
frames, 330
images, 235-238
bottom alignment, 237
left alignment, 237
middle alignment, 236
right alignment, 238
top alignment, 235
language-specific markup, 809
marquees, 258
page layout listing, 295
rows, 205
tables, 199
text, 376, 952
baseline, 952
bottom, 952
justification, 952
left, 156
middle, 952
<P> tag, 159
sub, 952
super, 952
text-bottom, 952
text-top, 952
top, 952
see also justification
ALINK attribute, <BODY> tag, 151
all value (clear attribute), 161
Allaire Corp. Web site, 694
allocating memory (Java), 570
Allow header, 99, 532
alt attribute
<IMG> tag, 235
<OBJECT> tag, 334
alt.lang.java newsgroup, 589
Alta Vista Web site, 481
Amaya Web site, 129, 963
American National Standards Institute (ANSI), 39, 840
American Standard Code for Information Interchange (ASCII), 840
ampersands (&), 57, 60
name/value pairs, 518
URL encoding, 520
anchors, 840
<A> tag, 223
e-mail links, 226-227
marking within documents, 228-232
connecting links, 230
enclosing headings, 229
placing tags before headings, 229
referencing filenames, 231
relative links, 224-225
<BASE> tag, 225-226
specifying URL, 223-224
W3C HTML 3.2 specification, 920
Andy's Photoshop tips Web site, 715
animated GIFs, 255-256, 668, 724-725
creating, 256
Alchemy Mindworks GIF Construction Set, 256, 724
GIF Animator, 724-725
GIFBuilder, 256, 668, 724
MultiGIF, 725
whirlGIF, 256
download time, 255
file size, 724
animated graphics, page layout, 455

IPR2024-00143
Apple EX1011 Page 507

animation, 723
Animator, 250-255
directory structure, 251
graphic files, 251
implementing animation, 252-253
location, 250
parameters, 254-255
sound format, 252
Macromedia Flash, 725
system requirements, 725
plug-ins, 663-668
Cosmo Player, 667
Shockwave, 665-666
Surround Video, 666
WIRL, 667
technologies available, 723
Animator, 250-255
directory structure, 251
graphic files, 251
implementing animation, 252-253
location, 250
parameters, 254-255
sound format, 252
Animator applet, 582-584
ANSI (American National Standards Institute), 39, 840
ANY content model, 59
AOL browser, 129-130
Apache Development Group Web site, 772
Apache Web server, 772-773
Apache Development Group, 772
Cygnus Web site, 773
mailing list, 773
technical support, 773
APIs, JavaScript, 546
Appearance node, 641
<APPLET> tag, 84, 248, 578-579, 872
syntax, 253-254
W3C HTML 3.2 specification, 923-925
applets (Java), 571, 840
Animator, 250-255, 582-584
directory structure, 251
graphic files, 251
implementing animation, 252-253
location, 250
parameters, 254-255
sound format, 252
<APPLET> tag, 248
Applet Viewer, 572-573
versus applications, 571-572
Backstage Designer, 747
browser compatibility, 571
Calendar, 584
creating, 574
class declaration, 574-575
paint method, 575-577
destroying, 576
example of, 577-578
HTML markup, 578
<APPLET> tag, 578-579
<PARAM> tag, 579
initializing, 576
J-Track, 587
Macromedia ImageMap PowerApplet, 585-586
marquees, 258
multimedia, 247
overriding, 576
scripts, 579-582
sources, 588
alt.lang.java, 589
Gamelan, 589
JARS, 588
JavaSoft, 588
starting, 576
stopping, 576
Ticker, 586-587
W3C HTML 3.2 specification, 923-925
applications
defined, 840
desktop publishing (exporting Web pages), 466
helper, 249, 846
connecting to, 250
viewers, 858
Java, 571-572
SGML
defining, 39-40
HTML as, 893-894
see also plug-ins
<AREA> tag, 269, 873
attributes
coords, 269-270
shape, 269
size, 269
Arena Web site, 963
ASCII (American Standard Code for Information Interchange), 840
ASP (active server pages), 683
Assets view (NetObjects Fusion), 754
asterisks (*), 60
Atkinson, Bill, 11-13
HyperCard, 12-13
limitations, 12
attaching scripts
to ActiveX controls, 610-613
to HTML controls, 596-598
to Layout control, 618-621
ATTLIST keyword, 62
attributes, 62-64, 74-75, 593
above, <LAYER> and <ILAYER> tags, 350
accept-charset, <FORM> tag, 803-804
accesskey, <LABEL> tag, 332
align
ActiveX controls, 261
<DIV> tag, 163
headings, 156
<HR> tag, 177
<IFRAME> tag, 330
<IMG> tag, 235-238
<MARQUEE> tag, 258
<OBJECT> tag, 334
<P> tag, 158

<TABLE> tag, 199-202
<TD> and <TH> tags, 212
<TR> tag, 205
ALINK, <BODY> tag, 151
alt
<IMG> tag, 235
<OBJECT> tag, 334
ATTLIST statements, 62
AUTOSTART, <EMBED> tag, 661
background
background, 951
background-attachment, 950
background-color, 950
background-image, 950
background-position, 950-951
background-repeat, 950
<BODY> tag, 149
<LAYER> and <ILAYER> tags, 350
order of, 951
behavior, <MARQUEE> tag, 257
below, <LAYER> and <ILAYER> tags, 350
bgcolor
<BODY> tag, 149-150
<LAYER> and <ILAYER> tags, 350
<MARQUEE> tag, 258
border, 954-955
border-color, 954
border-style, 955
border-width, 954
<FRAMESET> tag, 340
<IMG> tag, 240
<OBJECT> tag, 334
<TABLE> tag, 203-204
bordercolor, <TABLE> tag, 204
CDATA keyword, 62
cellpadding, <TABLE> tag, 203-204
cellspacing, <TABLE> tag, 203-204
class
<BUTTON> tag, 332
<FIELDSET> tag, 333
JavaScript style sheets, 403
<LABEL> tag, 331
classid, <OBJECT> tag, 248, 333
clear, <BR> tag, 159
clip, <LAYER> and <ILAYER> tags, 350
codebase
ActiveX controls, 261
<OBJECT> tag, 333
codetype
ActiveX controls, 261
<OBJECT> tag, 248, 334
color, <FONT> tag, 85, 175
cols, <FRAMESET> tag, 340
colspan, <TD> and <TH> tags, 209
compact, 192
content, <META> tag, 79
coords, <AREA> tag, 269-270
data, <OBJECT> tag, 333
declare, <OBJECT> tag, 333
default, 74
default value specifications, 63-64
defined, 23, 840
deprecated features, 64
marked sections, 64
dir, 808
<BUTTON> tag, 332
<FIELDSET> tag, 333
<LABEL> tag, 332
direction, <MARQUEE> tag, 258
disabled
<BUTTON> tag, 332
<LABEL> tag, 332
dynsrc
<IMG> tag, 256-257
platform dependence, 816
enctype, <FORM> tag, 277
face, <FONT> tag, 173
#FIXED keyword, 64
fonts, 948
font, 949
font-family, 948
font-size, 949
font-style, 948
font-variant, 948
font-weight, 948-949
order of, 949
for, <LABEL> tag, 331
frameborder
<FRAME> tag, 342
<FRAMESET> tag, 340
<IFRAME> tag, 329
generic
class, 326
id, 326
style, 326
grouping, style sheets, 381
height
<HR> tag, 177
<IFRAME> tag, 330
<IMG> tag, 239
<MARQUEE> tag, 258
<OBJECT> tag, 334
<TD> and <TH> tags, 209
hppt-equiv, <META> tag, 146
HREF, <BASE> tag, 76, 143
href
<A> tag, 223-224
<LINK> tag, 145, 325
hspace
<IMG> tag, 240
<MARQUEE> tag, 258
<OBJECT> tag, 334
http-equiv, <META> tag, 78
id, 414
<BUTTON> tag, 332
<FIELDSET> tag, 333
<LABEL> tag, 331
<OBJECT> tag, 333

IPR2024-00143
Apple EX1011 Page 508

ID keyword, 63
ids, JavaScript style sheets, 403
#IMPLIED keyword, 64
ismap, <IMG> tag, 241
lang, 806
<BUTTON> tag, 332
<FIELDSET> tag, 333
<LABEL> tag, 332
language, <SCRIPT> tag, 327, 593
layout style, 376-378
margin-left, 376
margin-right, 376
margin-top, 376
text-align, 376
text-indent, 376
left, <LAYER> and <ILAYER> tags, 351
length, 74
LINK, <BODY> tag, 151
loop, <MARQUEE> tag, 258
marginheight
<FRAME> tag, 342
<IFRAME> tag, 329
margins, 953
marginwidth
<FRAME> tag, 342
<IFRAME> tag, 329
maxlength, text fields, 280
media
<LINK> tag, 325
<STYLE> tag, 326
method, <FORM> tag, 276
name
<A> tag, 228
<FRAME> tag, 342
<IFRAME> tag, 329
<LAYER> and <ILAYER> tags, 351
<LINK> tag, 145
<META> tag, 146-147
<OBJECT> tag, 334
<PARAM> tag, 335
NAME keyword, 62
NMTOKEN keyword, 62
noresize, <FRAME> tag, 342
noshade, <HR> tag, 177
nowrap, <TD> and <TH> tags, 212
NUMBER keyword, 63
onBlur, <LABEL> tag, 332
onClick, <LABEL> tag, 332
onFocus, <LABEL> tag, 332
padding, 953-954
prompt, <ISINDEX> tag, 144
rel
<A> tag, 227
<LINK> tag, 145, 325
#REQUIRED keyword, 64
rev
<A> tag, 227
<LINK> tag, 145, 325
rows, <FRAMESET> tag, 341
rowspan, <TD> and <TH> tags, 209
SCRIPT element, 80
TYPE, 80
scrollamount, <MARQUEE> tag, 258
scrolldelay, <MARQUEE> tag, 258
scrolling
<FRAME> tag, 342
<IFRAME> tag, 330
shape, <AREA> tag, 269
shapes, <OBJECT> tag, 334
size
<AREA> tag, 269
<BASEFONT> tag, 84
<FONT> tag, 85, 173-175
text fields, 280
specifying, 74-75
src
<FRAME> tag, 342
<IFRAME> tag, 329
<IMG> tag, 235
<SCRIPT> tag, 327
standby, <OBJECT> tag, 334
stating values, 75
style
<BUTTON> tag, 332
<FIELDSET> tag, 333
<LABEL> tag, 331
JavaScript style sheets, 403
tabindex, <BUTTON> tag, 332
target, <A> tag, 226, 343-344
text
<BODY> tag, 150-151
letter-spacing, 951
line-height, 953
text-align, 952
text-decoration, 951
text-indent, 952
text-transform, 952
vertical-align, 952
word-spacing, 951
text boxes
clear, 955
float, 955
height, 955
width, 955
text style, 371-376
embedded style sheets, 374-376
font size, 372
font-family, 371
font-style, 373
font-weight, 373
line-height, 374
text-decoration, 373
title
<A> tag, 227-228
<BUTTON> tag, 332
<FIELDSET> tag, 333
<LABEL> tag, 332
<LINK> tag, 145, 325
<OBJECT> tag, 334
<STYLE> tag, 326
top, <LAYER> and <ILAYER> tags, 351
type
<LINK> tag, 325
<OBJECT> tag, 333
<SCRIPT> tag, 327

<STYLE> tag, 81, 326
<UL> tag, 190-192
usemap
<IMG> tag, 242, 269
<OBJECT> tag, 334
valign
<TD> and <TH> tags, 212
<TR> tag, 205
value
<LI> tag, 192
<PARAM> tag, 335
text fields, 280
valuetype, <PARAM> tag, 335
visibility, <LAYER> and <ILAYER> tags, 351
VLINK, <BODY> tag, 152
vspace
<IMG> tag, 240
<MARQUEE> tag, 258
<OBJECT> tag, 334
width
<HR> tag, 177
<IFRAME> tag, 330
<IMG> tag, 239
<LAYER> and <ILAYER> tags, 351
<MARQUEE> tag, 258
<OBJECT> tag, 334
<PRE> tag, 161
<TABLE> tag, 199-202
<TD> and <TH> tags, 208
XML, 785
zindex, <LAYER> and <ILAYER> tags, 351
audio
Animator, 250-255
directory structure, 251
implementing, 252-253
location, 250
parameters, 254-255
sound format, 252
background, 257
loading, 457
plug-ins, 659-663
QuickTime, 854
RealAudio, 249, 854
AUTH TYPE environment variable, 540
authentication, 840
Authorization header, 532
Authorization header field, 95
AUTOSTART attribute, <EMBED> tag, 661

B

<B> tag, 84, 171, 867
BackColor property (Label control), 608
background attribute, 951
<BODY> tag, 149
<LAYER> and <ILAYER> tags, 350
background-attachment attribute, 950
background-color attribute, 950
background-image attribute, 950
background-position attribute, 950-951
background-repeat attribute, 950
backgroundImage property, JavaScript style sheets, 409
backgrounds
attached versus fixed, 950
audio, 257
color, 950
BGCOLOR attribute, 149-150
marquees, 258
nested tables, 464
WebTV, 133
CSS, 379-380
color, 379
graphics, 379
splashes, 379
images, 242-243, 950
BACKGROUND attribute, 149
placement, 950
repeating, 950
JavaScript style sheets, 409
page layout, 456
Backstage, 730, 742-751
add-ons, 751
Administrator, 750
components, 742
Designer, 745-748
ActiveX controls, 747
applets, 747
formatting errors, 746
HTML validation, 749
images, 747
plug-ins, 748
script editor, 749
source-edit mode, 748
text capabilities, 746-747
file compatibility, 742
frames, 746
Manager, 742-745
maintaining links, 744-745
project elements, 743-744
remote administration, 743
objects, 742, 750
Server, 749-751
requirements, 750
strengths, 742
Backstage Internet Studio Web site, 966
bandwidth, 840
Bandwidth Buster, 704-705
compression, 704
downloading, 704
interface, 705
LOW SRC images, 704
text-only sites, 705
<BASE> tag, 76, 143-144, 863
HREF attribute, 76, 143
relative links, 225-226
W3C HTML 3.2 specification, 897

IPR2024-00143
Apple EX1011 Page 509

<BASEFONT> tag, 84-85, 869
size attribute, 84, 175
W3C HTML 3.2 specification, 926
BASESET keyword, 42
Basic Multilingual Plane (BMP), 799
BBEdit, 695
extensions, 695
Web site, 695, 964
BDO (BiDirectional Override) element, 808
Beginner's Guide to JavaScript Web site, 563
behavior attribute, <MARQUEE> tag, 257
below attribute, <LAYER> and <ILAYER> tags, 350
Berners-Lee, Tim, 13-14
BGCOLOR attribute, <BODY> tag, 149-150
bgcolor attribute
<LAYER> and <ILAYER> tags, 350
<MARQUEE> tag, 258
<BGSOUND> tag, 257, 871
BiDirectional Override element, 808
bidirectional text, 807
<BIG> tag, 84, 176, 868
binary numbers, hex notation, 846
bit, 840
bit combinations, 795
bitmap programs, 712-719
Adobe Photoshop, 713-715
features, 713-714
filters, 714
resource sites, 714-715
system requirements, 715
weaknesses, 714
advantages, 712
ColorWorks, 717
features, 717
multithreading, 717
SPG Web site, 717
system requirements, 717
common features, 713
Corel Photo-Paint, 718-719
features, 719
system requirements, 719
disadvantages, 712
editing programs, 712
Fractal Painter, 716
features, 716
system requirements, 716
vector import/editing, 716
GIMP, 717
system requirements, 717
Image Composer, 718, 741
features, 718
sprites, 718
system requirements, 718
Macromedia xRes, 718
paint programs, 712
PaintShop Pro, 715-716
download site, 715
features, 715
strengths, 715
system requirements, 716
weaknesses, 715
bitmaps, 841
dynamic graphics, 420-423
textures (VRML), 641
bits per second, 841
Bitstream Web site, 810
Black Diamond Web site, 666
<BLINK> tag (HTML), 869
block elements, 82, 841
ADDRESS, 82
BLOCKQUOTE, 82
CENTER, 82
DIV, 82
FORM, 82
format properties, 407-408
align, 408
borders, 407-408
clear, 408
height, 408
margins, 407
padding, 407
width, 408
Hn, 82
HR, 82
Lists, 82
P, 82
PRE, 82
SPAN, 82
TABLE, 82
Image Composer, 741
W3C HTML 3.2 specification, 902
block keyword, display attribute, 956
<BLOCKQUOTE> tag, 82, 166-167, 869
W3C HTML 3.2 specification, 908
BMP (Basic Multilingual Plane), 799
body (documents), 841
BODY element, 81-86, 148-152
block formatting elements, 82
ADDRESS, 82
BLOCKQUOTE, 82
CENTER, 82
DIV, 82
FORM, 82
Hn, 82
HR, 82
Lists, 82
P, 82
PRE, 82
SPAN, 82
TABLE, 82
tags, 82
character formatting elements, 83
logical markup, 83
physical markup, 83-84
formatting elements, 84-86
A, 84
APPLET, 84
BASEFONT, 84-85
BR, 84-85
FONT, 84-86
FORM, 84
IMG, 84
MAP, 84
OBJECT, 84

<BODY> tag, 148-152, 862
attributes
alink, 151
background, 149
bgcolor, 149-150
link, 151
text, 150-151
vlink, 152
W3C HTML 3.2 specification, 899-900
boldface type, 948
<B> tag, 171
<CAPTION> tag, 204
definition lists, 193
<STRONG> tag, 164
book cover style sheet, 397-398
bookmarks, 841
internal, 229-232
connecting links, 230
enclosing headings, 229
placing tags before headings, 229
referencing filenames, 231
border attribute, 955
<FRAMESET> tag, 340
<IMG> tag, 240
<OBJECT> tag, 334
<TABLE> tag, 203-204
border-bottom attribute, 954
border-bottom-width attribute, 954
border-color attribute, 954
border-left attribute, 954
border-left-width attribute, 954
border-right attribute, 954
border-right-width attribute, 954
border-style attribute, 955
border-top attribute, 954
border-top-width attribute, 954
border-width attribute, 954
bordercolor attribute, <TABLE> tag, 204
borders
frames, 329
images, 240
JavaScript style sheet properties, 407-408
text boxes (attributes), 954-955
border-color, 954
border-style, 955
border-width, 954
bots, 854
FrontPage, 738
capabilities, 738
inserting, 738
Save Results, 738
bounding boxes (text), 953
border attributes, 954-955
border-color, 954
border-style, 955
border-width, 954
clear attribute, 955
float attribute, 955
height attribute, 955
margin attributes, 953
padding attributes, 953-954
width attribute, 955
bps (bits per second), 841
**
 tag, 84-85, 159-161, 868**
<CODE> tag and, 168
ordered lists, 181-182
W3C HTML 3.2 specification, 926
browsers, 28-32
address style, 164
Amaya, 129
AOL, 129-130
authentication, 95
block quotations, 166
bookmarks, 841
caches, 841
client-side image maps, 271
communicating with CGI programs (HTTP headers), 524-525
content types, 247
controlling (JavaScript scripts), 561
CSS support, 454
defined, 841
face attribute (<FONT> tag), 173
file extensions, 247
formatting (logical paragraphs), 158
frame support, 338
heading levels, 156
history, 117-118
horizontal lines, 177
HotJava, add-ons, 259
Internet Explorer, 32, 125-128, 847
4.0 version, 125
ActiveX controls, 260-262
background sounds, 257
extensions, 128
FrontPad, 125
plug-ins, 259-260
scrolling marquees, 257-258
versions, 32
versus Navigator, 118-121
video, 256-257
Java applets support, 571
JavaScript support, 546-547
style sheets, 401
layer support, 348
<LI> tag attributes, 192
logical styles, 170
Lynx, 130, 850
monospaced text, 172
Mosaic, 28-29, 118, 130
Mozilla, 29-31, 850
HTML extensions, 30-31
multicolumn directory lists, 194
Navigator, 122-124
ActiveX controls, 260-262
extensions, 123-124
versus Internet Explorer, 118-121
plug-ins, 259-260
versions, 31
NCSA, 28

IPR2024-00143
Apple EX1011 Page 510

Netscape, 851
ordered lists, 181
starting positions, 189
plug-ins, 652-654
animation, 663-668
audio, 659-663
business, 676-677
communications/ conferencing, 672-674
external, 668-669
image, 674-676
integrating, 654
multilingual support, 669-672
video, 654-659
Pocket Internet Explorer, 130-131
preferences (table appearance), 203
providing substitute content, 820
SCRIPT element support, 80
STYLE element support, 81
substitute content, 243-244
tables
alignment, 200
preformatted text, 162
text color, 175
Viola, 28, 858
Wave, 131
Web sites, 962-964
Amaya, 963
Arena, 963
Browserwatch, 962
Cyberdog, 963
Galahad, 963
Internet Explorer, 963
Lynx, 963
NCSA Mosaic, 964
Netscape, 962
Opera, 964
SlipKnot, 964
WebTV, 132-133
background colors, 133
designing pages for, 132
WebTV Network site, 132
width attribute, 161
Browserwatch Web site, 962
browsing, defined, 222
bubbling (events), 423
buffers
images, 240
marquees, 258
built-in ActiveX controls, 606-607
bullets, unordered lists, 190
Bush, Vannevar, 8-9, 109
memex, 8-9, 109
business pages, style sheets, 386-387
business plug-ins, 676-677
Button control, 603
<BUTTON> tag, 332
attributes
class, 332
dir, 332
disabled, 332
id, 332
lang, 332
style, 332
tabindex, 332
title, 332
buttons (forms), 332
byte, 841
byte code verification (Java), 570

C

C, 841
C++, 841
c|net Web site, 706
cable-modem, 841
caches, 841
CAI (Computer Associates International), 686-687
ICE database, 686
Jasmine database, 686
Unicenter management systems Web site, 686
Calendar applet, 584
CAPACITY section (HTML 4.0 declaration), 51
capitalization, text-transform attribute, 952
Caption property (Label control), 609
<CAPTION> tag, 204, 877
W3C HTML 3.2 specification, 911
see also style sheets
cascading style sheets, see CSS
case sensitivity, entity names, 56
CDATA
content model, 59
keyword, 62
cellpadding attribute, <TABLE> tag, 203-204
cells (tables), 203, 841
aligning content, 212
attributes, 203
content, 212
precedence, 213-214
size, 208-211
data, 208
defining, 206
header, 207
cellspacing attribute, <TABLE> tag, 203-204
center attribute, table alignment, 200
<CENTER> tag, 82, 869
versus <DIV> tag, 163
W3C HTML 3.2 specification, 907
Centura, 685-686
Web Developer
requirements, 686
security, 686
Web site, 685
CERN (European Laboratory for Particle Physics), 112, 265, 844
Web site, 112
httpd Web server, 773
CERN servers, MAP files, 266-267
circle keyword, 266
default keyword, 266

polygon keyword, 266
rectangle keyword, 266
CGI (Common Gateway Interface) programs
activating, 521
calling, 514
databases, 680
decoding data, 521-524
CONTENT LENGTH environment variable, 522
converting special characters, 522
listing, 523-524
QUERY STRING environment variable, 522
REQUEST METHOD environment variable, 521
splitting name/value pairs, 522
environment variables, 537-541
AUTH TYPE, 540
CONTENT LENGTH, 522
CONTENT-LENGTH, 541
CONTENT-TYPE, 541
GATEWAY INTERFACE, 537
HTTP REQUEST METHOD, 538
PATH, 539
PATH INFO, 539
PATH TRANSLATED, 539
QUERY STRING, 522, 540
REMOTE ADDR, 540
REMOTE HOST, 540
REMOTE IDENT, 541
REMOTE USER, 540
request header-based, 538
REQUEST METHOD, 521
SCRIPT FILENAME, 539
SCRIPT NAME, 539
SERVER ADMIN, 537
SERVER NAME, 538
SERVER PORT, 538
SERVER PROTOCOL, 538
SERVER SOFTWARE, 538
server-based, 537
GET method, 516
encoded data listing, 517
registration form listing, 516-517
HTML and, 514-515
name/value pairs, 518
ampersands (&), 518
name field, 518
passing to servers, 518
value field, 518
paACTION field, 518
passing data to, 515
<FORM> tag, 514-515
PATH INFO variable, 515
path information, 518-519
versus scripting, 80
searching with, <ISINDEX> tag, 144-145
sending data to, HTTP headers, 524-525
URL encoding, 519-521
percent signs (%), 519
reserved characters, 519-521
resources, 521
steps, 521
Web site functions, 514
character encoding, 794-799, 841
7-bit ASCII, 795-796
control characters, 795
restrictions, 795
8-bit encodings, 796-797
control characters, 796
ISO 8859 standards, 796
16-bit encodings, 797-798
backward compatibility, 797
code switching, 797
bit combinations, 795
versus character sets, 795
external
specifying, 802-803
versus document character set, 801
ISO 10646, 798-799
BMP (Basic Multilingual Plane), 799
problems, 805-806
Unicode, 857
URLs, 105
character formatting elements, 83
logical markup, 83
CITE, 83
CODE, 83
DFN, 83
EM, 83
KBD, 83
Q, 83
SAMP, 83
STRONG, 83
VAR, 83
physical markup, 83-84
B, 84
BIG, 84
I, 84
SMALL, 84
STRIKE, 84
SUB, 84
SUP, 84
TT, 84
U, 84
character sets, 41-42, 842
versus character encoding, 795, 801
defining, 42
document, 800-802
Unicode advantages, 801-802
external character encoding, 43

IPR2024-00143
Apple EX1011 Page 511

HTML 4, 44-50
HTML Internationalization, 43
problems, 805-806
characters
glyphs, 845
kerning, 849
special (list), 882-889
CHARSET keyword, 42
charset parameter, 802
CHARSET section (HTML 4.0 declaration), 41-50
charsets (MIME), 799
Checkbox control, 603, 606
checkboxes, 281
children, grouping (VRML), 638
chunking, 842
citations, see sources
<CITE> tag, 83, 167, 867
class attribute, 326
<BUTTON> tag, 332
<FIELDSET> tag, 333
JavaScript style sheets, 403
<LABEL> tag, 331
classes, 842
assigning to tags, 381-382
declaring, 574-575
inheritance, 847
JavaScript, 549
style sheets, 401, 946
separating, 946
classid attribute, <OBJECT> tag, 248, 333
clear attribute

 tag, 159
floating images, 161
text boxes, 955
clear property, JavaScript style sheets, 408
client error response codes, 526
client-pull, 147
client-side
components, 606
defined, 842
image maps, 241, 264, 268-271
advantages, 268
<AREA> tag, 269
browser support, 271
enabling, 269
<MAP> tag, 269-271
clients
defined, 842
fat, 844
pull, 854
thin, 851, 857
clip attribute, <LAYER> and <ILAYER> tags, 350
clipping rectangles, 357
listing, 357
CMYK color model, 447
code (program)
<CODE> tag, 167-168
<KBD> tag, 169
<SAMP> tag, 169
code pages, 842
<CODE> tag, 83, 167-168, 867
codebase attribute
<APPLET> tag, 579
<OBJECT> tag, 333
ActiveX controls, 261
CodeBase property, 614
codetype attribute
<OBJECT> tag, 248, 334
ActiveX controls, 261
collaborative browsing, 122, 674
color
active links, 151
background, 950
background-color attribute, 950
BGCOLOR attribute, 149-150
CSS, 379
marquees, 258
nested tables, 464
WebTV, 133
borders, 954
defining, RGB hexadecimal triplet, 175
designating, 447
interfaces, 446-448
JavaScript style sheets, 409
links, 224
LINK attribute, 151
selecting, 447
text, 150-151
visited links, 152
Web palette, 447
color attribute, <FONT> tag, 85, 175
color models, 447
CMYK, 447
RGB, 447
color property, JavaScript style sheets, 409
colors
naming, 175
table cell borders, 204
text, 176
ColorWorks, 717
features, 717
multithreading, 717
SPG Web site, 717
system requirements, 717
cols attribute, <FRAMESET> tag, 340
colspan attribute, <TD> and <TH> tags, 209
columns
classic layout example, 296-297
spanning (listing), 209-210
COM (component object model), 592
combobox control, 606
commandbutton control, 606
commas (,), 60
selectors, 946
COMMENT element, 73
comment tags (<!-- -->), 40
nested lists, 189
<COMMENT> tag, 862
Common Gateway Interface programs, see CGI programs

communications plug-ins, 672-674
Communicator 4.0, 122
collaborative browsing, 122
compact attribute, 192
compilers (Java), 570
component object model (COM), 592
Composer, 122, 697-698, 965
browser integration, 697
capabilities, 698
editing screen, 697
Web site, 965
compression, 842
Computer Associates International, see CAI
computers
changing roles, 827-829
HTML and, 827
HPCs, 116-117
connections, 117
operating systems, 117
laptops/notebooks, 116
connections, 116
monitors, 116
NCs, 116
monitors, 116
nodes, 851
PCs, 115
conditional GETs (HTTP), 94
Conference software, 674
conferencing plug-ins, 672-674
connectionless protocols, 90
connections
external media, 250
HPCs, 117
HTTP, 91-93
HTTP-NG, 102
laptops/notebooks, 116
PCs, 115
site design considerations, 114
containers, 842
content (Web pages), 842
CONTENT attribute
META element, 79
CONTENT LENGTH environment variable, 522
content models
groups, 59-61
ampersands (&), 60
asterisks (*), 60
commas (,), 60
connectors, 60
#PCDATA keyword, 61
plus signs (+), 60
question marks (?), 60
vertical bars (\|), 60
keywords, 59
ANY, 59
CDATA, 59
EMPTY, 59
RCDATA, 59
content types, 247
Content-Encoding header, 99, 532
Content-Language header, 532
CONTENT-LENGTH environment variable, 541
Content-Length header, 99, 532, 536
Content-Transfer-Encoding header, 532
CONTENT-TYPE environment variable, 541
Content-Type header, 100, 532, 536
Content-Type response header, 525
contextual selectors, 946
Control Pad (ActiveX), 607-608
Script Wizard, 610-613
controlling
browsers, JavaScript scripts, 561
Java applets, 579-582
controls
ActiveX, 260-262, 593, 606
align attribute, 261
Backstage Designer, 747
built-in controls, 606-607
class IDs, 261
codebase attribute, 261
codetype attribute, 261
Control Pad, 607-608
data control objects, 426
including, 261
making available to users, 614
Microsoft HTML Layout control, 615-616
multimedia, 249
navigation, 477
properties, 608-610
scripts, 610-613
third-party controls, 607
interface, 442-446
hardware elements, 442
selecting, 445-446
software elements, 443
intrinsic HTML, 603
attaching scripts to, 596-598
page layout, 457
cookies, 842
coords attribute, <AREA> tag, 269-270
copyrights, 842
CorelDRAW!, 720-721
features, 720
system requirements, 721
Web site, 720
Corel Photo-Paint, 718-719
features, 719
system requirements, 719
Web site, 718
Corel Xara, 722
Corporate Presence wizard (Explorer), 733
corporate Web sites, 487-488
Cosmo Player plug-in, 667
CosmoPlayer VRML viewer, 632
Cougar, see HTML 4
Crunch!, 705-706
Web site, 705

IPR2024-00143
Apple EX1011 Page 512

CSS (cascading style sheets), 323-326, 386, 841
backgrounds, 379-380
color, 379
graphics, 379
splashes, 379
browser support, 454
Cascading Style Sheets 1 (CSS1), 946
combining, 370
cascading order, 370
DIV element, 326
embedded, 366-368
text style attributes, 374-376
examples of
business page, 386-387
employee newsletter, 388-390
Kid's News, 395-397
online book cover, 397-398
party invitation, 394-395
price list, 390-393
grouping, 381
assigning classes, 381-382
attributes, 381
tags, 381
implementation, 324
inline, 366-368
layout, 454-455
layout style attributes, 376-378
margin-left, 376
margin-right, 376
margin-top, 376
text-align, 376
text-indent, 376
LINK element, 325
linked, 366, 369, 390
navigation, 475
SPAN element, 326
style control, 364-366
fonts, 365
STYLE element, 325-326
text style attributes, 371-376
font size, 372
font-family, 371
font-style, 373
font-weight, 373
line-height, 374
text-decoration, 373
W3C recommendation, 961
see also style sheets
.css file extension, 369, 947
CU-SeeMe, 674
curly brackets ({}), style sheets, 946
cursive fonts, 371, 948
Customer Support Web template (Explorer), 733
customizing ordered lists, 184
Cyberbit font, 810
Cyberdog Web site, 963
CyberMad Web site, 486, 499-509
animation, 508
cover page, 508
interface design, 499-503
Hollywood Geek Machine, 499
Hype Monster, 501
navigation, 507-508
links, 508
navigation menu, 507
Table of Contents page, 508
page layout, 503-507
tables, 504-505
Cygnus Web site, 773

D

data attribute, <OBJECT> tag, 333
data cells, 208
Database Connector Wizard, 736
databases (Web site), 680
Access 97, 682-683
ASP, 683
dynamic approach, 682
IDC/HTX, 682-683
links, 682
static approach, 682
wizards, 683
Backstage Server add-on, 750
CAI, 686-687
ICE, 686
Jasmine, 686
Centura, 685-686
Web Developer, 686
CGI, 680
defined, 843
IBM DB/2, Net.Data, 685
Informix, 687
Lotus Approach 97, 681-682
NetObjects Fusion and, 753
Oracle, 684-685
Developer/2000 report writing module, 684
problems, 680
queries, Backstage Manager, 743
SQL Server, 684
publishing technologies, 684
Web Assistant, 684
Sybase, 683-684
NetImpact Dynamo, 683
web.sql, 683
Date header, 98, 533-535
DBMS, Web servers, 763
DCOM (Distributed COM), 593
<DD> tag, 192, 867
Debabelizer utility, 722-723
features, 723
SuperPalette, 723
system requirements, 723
declarations, 843
declare attribute, <OBJECT> tag, 333
declaring
DOCTYPEs, 73
Java classes, 574-575
XML elements, 787-788
XML entities, 788

DEF keyword, VRML, 640
defining
event handlers, 595
functions, 550
JavaScript objects, 547
definition lists
<DL> tag, 192-194
HTML 3.2 W3C specification, 905
definitions, <DFN> tag, 167
delimiters, HTML 4.0 declaration, 53
deprecated features, 64
marked sections, 64
descriptive markup, 36-39
DESCSET keyword, 42
Designer (Backstage), 745-748
ActiveX controls, 747
applets, 747
formatting errors, 746
HTML validation, 749
images, 747
plug-ins, 748
script editor, 749
source-edit mode, 748
text capabilities, 746-747
designing
forms, HTML Layout control, 615-616
interfaces, 436-439
audience, 440-441
color, 446-448
content, 441
controls, 442-446
criteria, 437
CyberMad Web site, 499-503
image styles, 448-449
look and feel of site, 441-442
media considerations, 438-439
Microsoft Sitebuilder site, 437
site goals, 439-440
themes, 439
Travelution Web site, 488-492
navigation systems, 470-472
clarity, 472
outlining site, 471
placement, 472
sites
connections, 114
CyberMad, 486, 499-509
Travelution, 486-498
Web pages, WebTV, 132
destroy method, 576
destroying Java applets, 576
Developer/2000 report writing module, 684
HTML forms, 684
HTML reports, 684
DFN element, 83
<DFN> tag, 83, 167, 867
versus <VAR> tag, 168
dialog boxes, defined, 843
dialup access, 843
digital technology, 830-831
digital watermarking, 859
dir attribute, 808
<BUTTON> tag, 332
<FIELDSET> tag, 333
<LABEL> tag, 332
<DIR> tag, 194, 866
W3C HTML 3.2 specification, 906
direction attribute, <MARQUEE> tag, 258
Director from Macromedia, 666
directories, Animator applet, 251
disabled attribute
<BUTTON> tag, 332
<LABEL> tag, 332
discussion groups, Backstage Manager, 743
Discussion Web wizard (Explorer), 733
display attribute, 956
block keyword, 956
inline keyword, 956
none keyword, 956
display property, JavaScript style sheets, 409
displaying
Java applets, 575-577
message boxes, 601
status bar messages, 560-561
displays, pixel sizes and, 113
Distributed COM (DCOM), 593
<DIV> tag, 82, 163, 367, 865
CSS, 326
versus <CENTER> tag, 163
W3C HTML 3.2 specification, 907
<DL> tag, 192-194, 867
W3C HTML 3.2 specification, 905
DNS (Domain Name Service), 843
Web servers, 763
<!DOCTYPE> tag, 73-74, 139
document character set, 800-802
versus external character encoding, 801
Unicode advantages, 801-802
Document object, 599
referencing, 601
Document Object Model, see Dynamic HTML
document type definitions, see DTDs
documents
body, 841
dividing (<DIV> tag), 163
formatting, 36, 845
heads, 846
horizontal lines, 176
internal bookmarks, 229-232
connecting links, 230
enclosing headings, 229
placing tags before headings, 229
referencing filenames, 231
layering, 849
layout, 849
physical markup, 853

IPR2024-00143
Apple EX1011 Page 513

structure, 36, 856
valid, 858
validators, 858
well-formed, 859
DOM (Document Object Model), see Dynamic HTML
Domain Name Service, see DNS
domain names, 843
downloading
animated GIF files, 255
graphics, reducing file size, 722-723
headers, 141
images, 306
layered pages, 349
Personal Web Server, 739
reducing time
Bandwidth Buster, 704-705
Crunch! utility, 705-706
drawString method, 577
<DT> tag, 192, 867
DTDs (document type definitions), 40, 843
deprecated features, 64
marked sections, 64
HTML, 65
Internet Explorer, 65
Netscape, 65
HTML 3.2
code listing, 929-940
HTML 4, 55-64
attributes, 62-64
elements, 58-61
entities, 55-58
XML
accessing, 786-787
advantages, 785-786
element declarations, 787-788
entity declarations, 788
Dynamic fonts, 411
dynamic graphics, 420-423
drag-and-drop listing, 421-422
ZINDEX values, 420
Dynamic HTML, 28, 335-336, 843
data-aware pages, 423-430
binding to records, 424-428
data source control, 424
dynamic presentation listing, 424-426
features, 424
generating/sorting table rows, 428, 428-430
events, 415-416
onmousemove, 423
future developments, 431
graphics, 420-423
drag-and-drop listing, 421-422
ZINDEX values, 420
object model, 414-415
text, 417-420
changing ranges, 420
changing styles, 420
code listing, 418
enabling, 420
Web sites, 415, 961
see also WBL example
dynamic pages, 843
dynamic text, 417-420
code listing, 418
enabling, 420
dynsrc attribute
<IMG> tag, 256-257
platform dependence, 816

E

e-mail
address tag, 165
links to, 226-227
e-zines, 844
CyberMad, 499-509
animation, 508
cover page, 508
interface design, 499-503
navigation, 507-508
page layout, 503-507
EasyStart Server, 657
editing Web pages
NetObjects Fusion, 753
PageDraw, 756
editors
bitmap, 712
FrontPad, 125
HTML, 692-699
Backstage Designer, 745-748
BBEdit, 695
Composer, 697-698
FrontPage, 734-737
HomeSite, 694-695
HotDog, 698-699
Navigator Gold, 696-697
PageMill, 692-694
resources, 706
source, 692
image maps, 271-274
LiveImage, 273-274
Map This!, 272-273
MapEdit, 271-272
Web sites, 964-966
Backstage Internet Studio, 966
BBEdit, 964
HomeSite, 964
HotDog, 965
HoTMetaL Pro, 965
Microsoft FrontPad, 965
Microsoft FrontPage 97, 965
Netscape Composer, 965
PageMill, 966
Webber, 965
Element object, 600
elements, 58-61, 72-74
attributes, 74-75
default, 74
length, 74
specifying, 74-75
stating values, 75
BASE, 76
HREF attribute, 76
BiDirectional Override, 808

block, 841
W3C HTML 3.2 specification, 902
block formatting, 82
ADDRESS, 82
BLOCKQUOTE, 82
CENTER, 82
DIV, 82
FORM, 82
Hn, 82
HR, 82
Lists, 82
P, 82
PRE, 82
SPAN, 82
TABLE, 82
tags, 82
BODY, 81-86
character formatting, 83
logical markup, 83
physical markup, 83-84
COMMENT, 73
containers, 842
content model groups, 59-61
ampersands (&), 60
asterisks (*), 60
commas (,), 60
connectors, 60
#PCDATA keyword, 61
plus signs (+), 60
question marks (?), 60
vertical bars (\|), 60
content model keywords, 59
ANY, 59
CDATA, 59
EMPTY, 59
RCDATA, 59
defined, 23, 843
DOCTYPE, 73-74
declaration, 73
font style, W3C HTML 3.2 specification, 913-914
FRAMESET, 76
HEAD, 75-81
ISINDEX, 77
LINK, 77
attributes, 77
META, 78-79
attributes, 78
expiration dates, 78
refreshing, 79
minimization indicators, 58
NOSCRIPT, 80
phrase, W3C HTML 3.2 specification, 914
SCRIPT, 79-80
attributes, 80
browser support, 80
disadvantages, 80
versus CGI, 80
structural
<BASE> tag, 143-144
<BODY> tag, 148-152
<!DOCTYPE> tag, 139
<HEAD> tag, 140-141
<HTML> tag, 139-140
<ISINDEX> tag, 144-145
<LINK> tag, 145-146
<META> tag, 146-148
<SCRIPT> tag, 152-153
<STYLE> tag, 148
<TITLE> tag, 141-143
STYLE, 81
browser support, 81
versus tags, 73
text-level, 857
W3C HTML 3.2 specification, 913
TITLE, 81
XML, declaring, 787-788
see also tags
Elements object, referencing, 603-604
EM element, 83
em spaces, 951
<EM> tag, 83, 164-165, 867
<EMBED> tag, 660, 872-873
plug-ins, 260
embedded lists, see nested lists
embedded objects, 843
embedded style sheets, 366-368, 947
format, 947
syntax, 366
text style attributes, 374-376
embedding
event handlers, 415
Java applets, 578
<APPLET> tag, 578-579
<PARAM> tag, 579
JavaScript scripts, 555-556
plug-ins, 260
VBScript scripts
<INPUT> tag, 596-598
<SCRIPT> tag, 593-596
VRML, 644-645
billiard table listing, 644-645
emphasis
options, 164
tags, 164
employee newsletters (style sheets), 388-390
EMPTY content model, 59
Empty Web template (Explorer), 733
Encoder (RealAudio), 662
encoding URLs, 105, 519-521
decoding data, 521-524
percent signs (%), 519
reserved characters, 519-521
resources, 521
steps, 521
encryption, 843
enctype attribute, <FORM> tag, 277
Engelbart, Doug, 10, 109
NLS, 10
entities (HTTP), 55-58, 99-101, 786, 844
Allow header field, 99
Content-Encoding header field, 99
Content-Length header field, 99
Content-Type header field, 100

IPR2024-00143
Apple EX1011 Page 514

declaring, 56
Entity-Body, 100-101
Content-Encoded field, 101
Content-Length field, 101
Content-Type field, 100
Expires header field, 100
general, 57-58
ampersand character (&), 57
versus parameter, 57-58
header fields, 99
Last-Modified header field, 100
parameter, 56
case sensitivity, 56
percent signs (%), 56
public identifiers, 56-57
XML, declaring, 788
Entity-Body file, 100-101
Content-Encoded field, 101
Content-Length field, 101
Content-Type field, 100
environment variables (CGI), 537-541
AUTH TYPE, 540
CONTENT-LENGTH, 541
CONTENT-TYPE, 541
GATEWAY INTERFACE, 537
CGI specification, 537
HTTP REQUEST METHOD, 538
PATH, 539
PATH INFO, 539
PATH TRANSLATED, 539
QUERY STRING, 522, 540
REMOTE ADDR, 540
REMOTE HOST, 540
REMOTE IDENT, 541
REMOTE USER, 540
request header-based, 538
REQUEST METHOD, 521-522
SCRIPT FILENAME, 539
SCRIPT NAME, 539
SERVER ADMIN, 537
SERVER NAME, 538
SERVER PORT, 538
SERVER PROTOCOL, 538
SERVER SOFTWARE, 538
server-based, 537
equals signs (=), 74
URL encoding, 520
Equilibrium Web site, 723
error-handling (Java), 570
errors (HTTP), Authorization, 95
European Laboratory for Particle Physics, see CERN
event handlers, 595, 844
defining, 595
embedding, 415
events, 637, 844
bubbling, 423
Dynamic HTML, 415-416
onmousemove, 423
intrinsic, 328
onBlur, 328
onChange, 328
onClick, 328
onFocus, 328
onLoad, 328
onMouseOut, 328
onMouseOver, 328
onSelect, 328
onSubmit, 328
onUnload, 328
Excite Web site, 481
executing scripts
automatically, 594-595
on events, 595
subroutines, 595-596
expandP( ) function, 420
expiration dates (pages), 78
Expires header, 100, 533
explicit labels, 332
Explorer (FrontPage), 731-734
Folder View, 732
Hyperlink View, 731
templates, 733
Customer Support Web, 733
Empty Web, 733
Normal Web, 733
Personal Web, 733
Project Web, 733
wizards, 733
Corporate Presence, 733
Discussion Web, 733
Import Web, 733
exporting Web pages, 466
extended identifiers, 806
registry, 806
extended locators, XML, 789
Extensible Markup Language, see XML
extensions (HTML), 844
advantages, 817-819
influencing standards, 818-819
knowledge of future standards, 818
use of new technology, 818
common, 820-821
disadvantages, 819
lack of support, 819
syntax changes, 819
Internet Explorer, 65, 128
multiple pages, 820
Netscape, 30-31, 65, 123-124, 416-417
problems, 31
reasons for creating, 814-815
source, 814
versus standards, 112-113, 820
horizontal rules listing, 112
workarounds, 822-823
frames, 822
multimedia files, 822-823
see also file extensions
external character encoding, 43
specifying, 802-803
charset parameter, 802

external entities, 786
external multimedia, 246, 249-250
connecting to, 250
helper applications, 249
RealAudio, 249
external plug-ins, 668-669
external style sheets, see linking, style sheets
extranets, 844

F

face attribute, <FONT> tag, 173
fancy fonts, 948
fantasy fonts, 371
FAQs
<CITE> tag, 167
definition lists, 192
fat clients, 844
FEATURES section, 54-55
LINK class, 55
MINIMIZE class, 54-55
OTHER class, 55
<FIELDSET> tag, 333
attributes
class, 333
dir, 333
id, 333
lang, 333
style, 333
title, 333
<FIG> tag, 270
file extensions, 247, 844
.css, 369, 947
.wrz, 632
file paths (URIs), 103
file scheme, 106
file transfer protocol, see ftp
files
accessing (Java), 570
animated GIF, 255-256
creating, 256
download time, 255
Entity-Body, 100-101
Content-Encoded field, 101
Content-Length field, 101
Content-Type field, 100
MAP, creating, 266-268
filters, Adobe Photoshop, 714
firewalls, 845
first-letter values, 946
first-line values (selectors), 946
#FIXED keyword, 64
float attribute, text boxes, 955
floating images, clear attribute, 161
font attribute, 949
font size attribute, 372
centimeters, 372
inches, 372
percentages, 372
pixels, 372
points, 372
<FONT> tag, 84-86, 173, 869
attributes
color, 85, 175
size, 85, 173-175
W3C HTML 3.2 specification, 925
font-family attributes, 371, 948
cursive, 371
family name, 948
fantasy, 371
generic family, 948
cursive, 948
fancy, 948
monospace, 948
sans-serif, 948
serif, 948
monospace, 371
sans serif, 371
serif, 371
font-size attribute, 949
absolute size, 949
length, 949
percentage value, 949
relative size, 949
font-style attribute, 373, 948
italic value, 948
normal value, 948
oblique value, 948
font-variant attribute, 948
normal value, 948
small-caps value, 948
font-weight attribute, 373, 948-949
bold value, 948
normal value, 948
fontFamily property, JavaScript style sheets, 404
fonts
attributes, 948
font, 949
font size, 372
font-family, 371, 948
font-size, 949
font-style, 373, 948
font-variant, 948
font-weight, 373, 948-949
order of, 949
CSS, 365
cursive, 371, 948
defined, 845
Dynamic, 411
fancy, 948
fantasy, 371
font style elements, W3C HTML 3.2 specification, 913-914
<FONTS> tag, 173
color attribute, 175
size attribute, 173-175
internationalization, 810-811
Cyberbit, 810
glyph servers, 810
OpenType, 810
italic values, 948
JavaScript style sheets, 404-405
fontFamily property, 404
fontSize property, 404
fontStyle property, 405

IPR2024-00143
Apple EX1011 Page 515

layout, 454
monospace, 371, 850, 948
<TT> tag, 172
monospaced, 850
<PRE> tag, 162
normal values, 948
oblique values, 948
proportional, 853
sans serif, 371, 948
serif, 371, 948
size
absolute, 949
centimeters, 372
inches, 372
length, 949
percentage value, 949
percentages, 372
pixels, 372
points, 372
relative, 949
fontSize property (JavaScript style sheets), 404
fontStyle property (JavaScript style sheets), 405
for attribute, <LABEL> tag, 331
ForeColor property (Label control), 609
form fields, W3C HTML 3.2 specification, 914
Form object, 600
referencing, 601-603
Form Page Wizard, 736
form-based file upload, ftp site, 943
<FORM> tag, 82-84, 276-277, 874
attributes
accept-charset, 803-804
enctype, 277
method, 276
ACTION field, 518, 521
CGI programs, 514-515
GET method, 276
POST method, 276-277
W3C HTML 3.2 specification, 908
format (documents), 845
formatting, 36
<BIG> tag, 176
by usage, 163-164
documents, <DIV> tag, 163
<DT> and <DD> tags, 193
headings, 156-157
horizontal lines (<HR> tag), 176
lists, 957
paragraphs, 158
*
 tag, 159-161*
<P> tag, 158-159
<PRE> tag, 161-162
<XMP> tag, 163
physical styles, 170
<B> tag, 171
<FONT> tag, 173-176
<I> tag, 171
<STRIKE> tag, 171-172
<SUB> tag, 172-173
<SUP> tag, 172-173
<TT> tag, 172
<U> tag, 171
<SMALL> tag, 176
text, 156
marquees, 257
formatting elements, 84-86
A, 84
APPLET, 84
BASEFONT, 84-85
SIZE attribute, 84
block formatting, 82
ADDRESS, 82
BLOCKQUOTE, 82
CENTER, 82
DIV, 82
FORM, 82
Hn, 82
HR, 82
Lists, 82
P, 82
PRE, 82
SPAN, 82
TABLE, 82
tags, 82
BR, 84-85
character formatting, 83
logical markup, 83
physical markup, 83-84
FONT, 84-86
COLOR attribute, 85
SIZE attribute, 85
FORM, 84
IMG, 84
MAP, 84
OBJECT, 84
forms, 331, 845
BUTTON element, 332
designing, HTML Layout control, 615-616
Developer/2000 report writing module, 684
elements within, 276
FIELDSET element, 333
<FORM> tag, 276-277
enctype attribute, 277
GET method, 276
method attribute, 276
POST method, 276-277
guest book example, 286-288
listing, 288
guidelines, 278
HTML 3.2 W3C specification, 908
internationalization, 803-804
accept-charset attribute, 803-804
input types, 278-283
checkboxes, 281
hidden fields, 281
image, 283
password fields, 280
radio buttons, 282
reset buttons, 283
selection boxes, 285-286
submit buttons, 282-283
text areas, 284
text fields, 279-280
LABEL element, 331-332
explicit labels, 332
implicit labels, 332

sending data to servers
decoding data, 521-524
encoded data listing, 517
GET method, 516
name/value pairs, 518
path information, 518-519
registration form listing, 516-517
URL encoding, 519-521
testing, POST method, 277
validating, JavaScript scripts, 557-559
Forwarded header, 533
Fractal Design Corporation Web site, 444
Fractal Expression, 720
Web site, 720
fractal images, 727
Fractint, 727
Winfract, 727
Xfractint, 728
Fractal Painter, 716
features, 716
system requirements, 716
vector import/editing, 716
Web site, 716
Fractint, 727
Frame objects, 599
nesting, 605
<FRAME> tag, 341-342, 881-882
attributes
frameborder, 342
marginheight, 342
marginwidth, 342
name, 342
noresize, 342
scrolling, 342
src, 342
listing, 341
frameborder attribute
<FRAME> tag, 342
<FRAMESET> tag, 340
<IFRAME> tag, 329
frames, 328-330, 338-339, 845
aligning, 330
borders, 329
browser support, 338
considerations, 338-339
creating, 339-344
<FRAME> tag, 341-342
<FRAMESET> tag, 340-341
HTML controls, referencing, 604-605
<IFRAME> tag, 329-330, 477
listing, 330
layout, 344-347
nesting frames, 346
simple layout, 344-346
loading URLs to, 226
navigation, 476-477
advantages, 476
disadvantages, 476
necessity, 339
pure HTML, 822
uses, 338
versus layers, 348
VRML, 645-646
listing, 645
Frames Wizard, 737
<FRAMESET> tag, 76, 340-341, 881
attributes
border, 340
cols, 340
frameborder, 340
rows, 341
FRC 2070 document, 794
From header, 95, 533
FrontPad, 125
Web site, 965
FrontPage, 730-741
add-ons, 739-741
Image Composer, 741
Personal Web Server, 739
Web Publishing Wizard, 740-741
Editor, 734-737
forms, 735
hyperlink creation, 735
interface, 735
templates, 736
wizards, 736-737
Explorer, 731-734
Folder View, 732
Hyperlink View, 731
templates, 733
wizards, 733
security, 733-734
administrator permissions, 734
author permissions, 734
end user permissions, 733
screening permissions, 734
To Do list, 738
Web bots, 738
capabilities, 738
inserting, 738
Save Results, 738
Web site, 965
ftp (file transfer protocol), 89, 844
sites
form-based file upload document, 943
HTML 2.0 definition, 943
HTML table model, 943
URLs, 106
Web servers, 763
Full Method request header, 530
GET method, 530-531
HTTP version, 532-534
method types, 530
Requested URL, 531
Full-Response (HTTP), 97
Date header field, 98
Location field, 99
Pragma header field, 98
Server field, 99
Status-Line, 97
Reason-Phrase, 97
Status-Code, 97
function characters, HTML 4.0 declaration, 52
functions
expandP( ), 420
InStr( ), 619
JavaScript, 550-551
Now( ), 604
see also methods

IPR2024-00143
Apple EX1011 Page 516

G

Galahad Web site, 963
Gamelan Web site, 562, 589
gateways (proxy), 854
general entities, 57-58
ampersand character (&), 57
versus parameter, 57-58
General Image Manipulation Program, see GIMP
Generalized Markup Language (GML), 39
generating random numbers, 559-560
generic attributes
class, 326
id, 326
style, 326
GET method (HTTP), 94, 276, 516
conditional GETs, 94
encoded data listing, 517
registration form listing, 516-517
request header, 529-531
proxy, 531
GIF (Graphics Interchange Format) , 668, 724-725, 845
animated files, 255-256
creating, 256
download time, 255
file size, 724
GIF Animator, 724-725
GIF Construction Set, 668, 724
GifBuilder, 668, 724
MultiGIF, 725
GIF Animator, 724-725
Web site, 724
GIF Construction Set, 668, 724
Web site, 724
GifBuilder, 668, 724
Web site, 256, 724
GIMP (General Image Manipulation Program), 717
system requirements, 717
Web site, 717
glossary lists, <DL> tag, 192-194
glyph servers, 810
Web sites, 810
glyphs, 845
GML (Generalized Markup Language), 39
Gopher
URLs, 107
Web servers, 763
gopher scheme, 107
gophers, 845
graphical browsers, see browsers
graphical user interface, 845
graphics
3-D
ray tracers, 727
see also VRML
animation, 723
animated GIFs, 724-725
layout, 455
Macromedia Flash, 725
technologies available, 723
bitmap, 841
dynamic, 420-423
drag-and-drop listing, 421-422
ZINDEX values, 420
HTML+, 24-25
layout, 455, 465-466
file sizes, 465
graphics-only programs, 465-466
pixels, 853
raster, 854
reducing file sizes (Debabelizer), 722-723
table size, 201
vector, 858
see also images
Graphics Interchange Format, see GIF
graphics packages
bitmap, 712-719
Adobe Photoshop, 713-715
advantages, 712
ColorWorks, 717
common features, 713
Corel Photo-Paint, 718-719
disadvantages, 712
editing programs, 712
Fractal Painter, 716
GIMP, 717
Image Composer, 718, 741
Macromedia xRes, 718
paint programs, 712
PaintShop Pro, 715-716
choosing, 711
hardware requirements, 711
vector, 711-712, 719-722
Adobe Illustrator, 721
advantages, 711, 719
common features, 720
CorelDRAW!, 720-721
Corel Xara, 722
disadvantages, 712
file sizes, 711
Fractal Expression, 720
Macromedia FreeHand, 721-722
resolution loss, 711
Group node, 638
grouping
children (VRML), 638
style sheets, 381
assigning classes, 381-382
attributes, 381
tags, 381
guest books
form example, 286-288
listing, 288
Missoula Rural Fire District example, 313-315
GUI (graphical user interface), 845
Gupta, see Centura

H

<H1> to <H6> tags, 156-157, 865
handheld personal computers, see HPCs
hardware
graphics packages, 711
interface elements, 442
HEAD element, 75-81, 140-148
<BASE> tag, 76, 143-144
HREF attribute, 76
<FRAMESET> tag, 76
<ISINDEX> tag, 77, 144-145
<LINK> tag, 77, 145-146
<META> tag, 78-79, 146-148
attributes, 78
expiration dates, 78
refreshing, 79
hppt-equiv attribute, 146
name attribute, 146-147
<SCRIPT> tag, 79-80
attributes, 80
browser support, 80
disadvantages, 80
versus CGI, 80
<STYLE> tag, 148
browser support, 81
<TITLE> tag, 81, 141-143
HEAD method (HTTP), 94
<HEAD> tag, 76, 140-141, 862
listing, 141
W3C HTML 3.2 specification, 895
header cells, 207
header fields
entity (Allow), 99
general
Date, 98
Pragma, 98
request, 95-96
Authorization, 95
From, 95
If-Modified-Since, 96
Referer, 96
User-Agent, 96
response
Content-Encoding, 99
Content-Length, 99
Content-Type, 100
Expires, 100
Last-Modified, 100
Location, 99
Server, 99
headers, 524-525, 846
Accept, 532
Accept-Charset, 532
Accept-Encoding, 532
Accept-Language, 532
Allow, 532
Authorization, 532
<BASE> tag, 143-144
Content-Encoding, 532
Content-Language, 532
Content-Length, 532
Content-Transfer-Encoding, 532
Content-Type, 532
Date, 533
determining, 524
downloading, 141
Expires, 533
Forwarded, 533
From, 533
<HEAD> tag, 140-141
If-Modified-Since, 533
<ISINDEX> tag, 144-145
prompt attribute, 144
Last-Modified, 533
level 0 HTML, 22
Link, 533
<LINK> tag, 145-146
href attribute, 145
name attribute, 145
rel attribute, 145
rev attribute, 145
title attribute, 145
Location, 533
<META> tag, 146-148
hppt-equiv attribute, 146
name attribute, 146-147
MIME-Version, 533
Orig-URL, 533
Pragma, 533
Public, 533
Referer, 533
request
environment variables based on, 538
Full Method, 530
Method, 529-530
response, 525, 534-537
Content-Length, 536
Content-Type, 525, 536
Date, 535
enclosed URL, 536-537
Last Modified, 536
Server, 536
Set-Cookie, 536
status codes, 525-529
Status Response line, 534
Retry-After, 533
Server, 534
<STYLE> tag, 148
Title, 534
<TITLE> tag, 141-143
URL-header, 534
User-Agent, 534
WWW-Authenticate, 534
headings
<BLOCKQUOTE> tag, 166
<H1> to <H6> tags, 156-157
typefaces, 157
W3C HTML 3.2 specification, 901
heads (document), 846
height attribute
<HR> tag, 177
<IFRAME> tag, 330
<IMG> tag, 239
<MARQUEE> tag, 258
<OBJECT> tag, 334
<TD> and <TH> tags, 209
text boxes, 955
height property, JavaScript style sheets, 408

IPR2024-00143
Apple EX1011 Page 517

helper applications, 249, 846
connecting to, 250
viewers, 858
see also plug-ins
hex (hexadecimal) notation, 846
Hidden control, 603
hidden fields, 281
hiding JavaScript scripts, 555
hierarchy, HTML object module, 599-600
hits, 846
Hn element, 82
home pages, creating from templates, 308-311
HomeSite, 694-695
browser-specific tags, 694
drawbacks, 695
interface, 694
Web site, 964
horizontal lines
<HR> tag, 176
HTML 3.2 W3C specification, 908-909
hosts, 846
HotDog, 698-699
FTP capabilities, 699
interface, 698
style sheet support, 699
Web site, 965
HotJava add-ons, 259
HoTMetaL Pro Web site, 965
hotspot control, 606
hotspots, 264, 846
HPCs (handheld personal computers), 116-117, 846
connections, 117
operating systems, 117
<HR> tag, 82, 176, 868
W3C HTML 3.2 specification, 908-909
HREF attribute, <BASE> tag, 76, 143
href attribute
<A> tag, 223-224
defining link type, 227
relative links, 224-225
<LINK> tag, 145, 325
hspace attribute
<IMG> tag, 240
<MARQUEE> tag, 258
<OBJECT> tag, 334
HTML (HyperText Markup Language), 847
CGI and, 514-515
controls, 603
attaching scripts to, 596-598
development, 18
SGML importance, 66
DTDs, 65
Internet Explorer, 65
Netscape, 65
evolution
computer role changes and, 827
digital technology, 830-831
mass-market access, 831
standardized format and, 831-833
user interfaces, 829-830
extensions, 844
advantages, 817-819
common, 820-821
disadvantages, 819
multiple pages, 820
Netscape, 416-417
versus pure HTML, 820
workarounds, 822-823
non-Web applications of, 834-835
object model, 598
Document object, 601
Elements object, 603-604
Form object, 601-603
hierarchy, 599-600
variables, 604
Window object, 601
programming with, 833
M. Casco Web site, 833
proposals, 27-28
style sheets, 27
pure, 814, 821
advantages, 816
disadvantages, 816-817
requirements for, 815
resources, 960-966
browser technology sites, 962-964
editors/tools, 964-966
HTML Guru site, 962
HTML Writers Guild site, 962
Microsoft Site Builder Network site, 960
Netscape DevEdge OnLine site, 960
newsgroups, 966
W3C site, 960-961
Web Design Group site, 962
Yahoo! site, 961
standards, versus extensions, 112-113
VBScript markup, 593
<INPUT> tag, 596-598
<SCRIPT> tag, 593-596
VRML and
communication, 646-647
differences, 636-637
similarities, 635-636
see also Dynamic HTML
HTML 0, 22-23
headers, 22
paragraphs, 22
typical page example, 23
HTML 1, 23-24
FORM element, 24
style elements, 24
HTML 2, 25-26
ftp site, 943
specification, 961
HTML 3, 26-27
style sheets, 26
HTML 3.2
applet syntax, 253-254
multimedia, 250
specification, 961
W3C specification, 892-942
A element, 920
ADDRESS element, 901
APPLET element, 923-925

BASE element, 897
BASEFONT element, 926
block elements, 902
BLOCKQUOTE element, 908
BODY element, 899-900
BR element, 926
CAPTION element, 911
CENTER element, 907
DIR element, 906
DIV element, 907
DL element, 905
document structure, 894
DTD, 929-940
FONT element, 925
font style elements, 913-914
FORM element, 908
form fields, 914
HEAD element, 895
headings, 901
HR element, 908-909
IMG element, 920-923
INPUT element, 915-918
ISINDEX element, 896
ISO Latin-1 character entities, 940-942
LINK element, 898-899
MAP element, 926-927
MENU element, 906
META element, 897
OL element, 904-905
P element, 903
phrase elements, 914
PRE element, 906-907
sample SGML open catalog, 927
SCRIPT element, 896
SELECT element, 918-919
SGML declaration, 927-929
STYLE element, 896
TABLE element, 909-913
TD element, 911
text-level elements, 913
TEXTAREA element, 919
TH element, 911
TITLE element, 895-896
TR element, 911
UL element, 903-904
HTML 4, 27, 320-321, 410-411, 842
cascading style sheets, 323-326
DIV element, 326
implementation, 324
LINK element, 325
SPAN element, 326
STYLE element, 325-326
character set, 44-50
character support, 335
DTD, 55-64
enhancements, 321
forms, 331
BUTTON element, 332
FIELDSET element, 333
LABEL element, 331-332
frames, 328-330
<IFRAME> tag, 329-330
generic attributes
class, 326
id, 326
style, 326
iframe element, 477
internationalization, 322, 335
intrinsic events, 328
onBlur, 328
onChange, 328
onClick, 328
onFocus, 328
onLoad, 328
onMouseOut, 328
onMouseOver, 328
onSelect, 328
onSubmit, 328
onUnload, 328
links, 322
MIME types, 335
OBJECT tag, 333-334
expansion, 247
PARAM element, 334-335
SGML declaration, 40-55
CAPACITY section, 51
character set declaration, 52
CHARSET section, 41-50
delimiters, 53
FEATURES section, 54-55
function characters, 52
naming rules, 53
quantity limits, 54
reserved names, 53
shunned characters, 51-52
SYNTAX section, 51-54
scripting, 326-328
SCRIPT element, 327
tables, 335, 456-457
W3C Web site, 961
HTML editors, 692-699
Backstage Designer, 745-748
ActiveX controls, 747
applets, 747
formatting errors, 746
HTML validation, 749
images, 747
plug-ins, 748
script editor, 749
source-edit mode, 748
text capabilities, 746-747
BBEdit, 695
extensions, 695
Composer, 697-698
browser integration, 697
capabilities, 698
editing screen, 697
FrontPage Editor, 734-737
forms, 735
hyperlink creation, 735
interface, 735
templates, 736
wizards, 736-737
HomeSite, 694-695
browser-specific tags, 694
drawbacks, 695
interface, 694

IPR2024-00143
Apple EX1011 Page 518

HotDog, 698-699
FTP capabilities, 699
interface, 698
style sheet support, 699
Navigator Gold, 696-697
features, 697
posting pages, 697
PageMill, 692-694
features, 693
hidden HTML option, 693
Macintosh version, 693
resources, 706
source, 692
HTML Guru Web site, 962
HTML Layout control, 615
control properties, changing, 617-618
controls, adding, 616
example of, 622, 625-627
forms, designing, 615-616
inserting, 621-622
scripts, attaching, 618-621
HTML Validator, 699-701
capabilities, 700
cost, 701
syntax error list, 700
tags required, 701
Web site, 700
HTML Writers Guild Web site, 962
HTML+, 24-25
<HTML> tag, 139-140, 862
advantages, 140
listing, 140
HTTP, 88, 91-92, 847
0.9, requests, 93
1.0
Authorization request header field, 95
From request header field, 95
GET method, 94
HEAD method, 94
If-Modified-Since request header field, 96
POST method, 94
Referer request header field, 96
request methods, 93
Request-Line, 93-94
Request-URI, 94-95
requests, 93
User-Agent request header field, 96
1.1, 101
pipelining, 101
connections, 91-93
entities, 99-101
Allow header field, 99
Content-Encoding header field, 99
Content-Length header field, 99
Content-Type header field, 100
Entity-Body, 100-101
Expires header field, 100
header fields, 99
Last-Modified header field, 100
headers, 524-525
Accept, 532
Accept-Charset, 532
Accept-Encoding, 532
Accept-Language, 532
Allow, 532
Authorization, 532
Content-Encoding, 532
Content-Language, 532
Content-Length, 532
Content-Transfer-Encoding, 532
Content-Type, 525, 532
Date, 533
determining, 524
Expires, 533
Forwarded, 533
From, 533
If-Modified-Since, 533
Last-Modified, 533
Link, 533
Location, 533
MIME-Version, 533
Orig-URL, 533
Pragma, 533
Public, 533
Referer, 533
response, 525-529, 534-537
Retry-After, 533
Server, 534
Title, 534
URL-header, 534
User-Agent, 534
WWW-Authenticate, 534
HTTP-NG, 102
connections, 102
requests, 91-93
headers, 846
responses, 91, 96-99
Date header field, 98
Full-Response, 97
Location field, 99
Pragma header field, 98
Server field, 99
Status-Line, 97
stateless protocols, 91
transactions supported, 91
URL syntax, 107
weaknesses, 101
Web servers, 763
Web site, 943
HTTP REQUEST METHOD environment variable, 538
http-equiv attribute
<META> tag, 78, 146
variables, 146-147
HyperCard, 12-13, 846
limitations, 12
hyperfiles, 11
hyperlinks, see links
hypermedia, 846
ShockWave, 855
HyperTalk, 12, 846
hypertext, 15-16, 846
defined, 222
dividing content, 15
history, 10-13
Atkinson, Bill, 11-13
HyperCard, 12-13
Nelson, Ted, 10-11
Xanadu, 11

links, 222
nodes, 222
providing author context, 16
providing reader context, 15-16
Hypertext Markup Language, see HTML
hyperworld, 11
hyphenation, language-specific markup, 809

I

<I> tag, 84, 171, 867
ordered lists, 184
IBM DB/2, Net.Data, 685
security, 685
Web macro, 685
ICE (OpenIngres/Internet Commerce Enabled) database, 686
Web site, 686
icons, defined, 847
id attribute, 326, 414
<BUTTON> tag, 332
<FIELDSET> tag, 333
<LABEL> tag, 331
<OBJECT> tag, 333
ID keyword, 63
IDC/HTX, 682-683
ids attribute, JavaScript style sheets, 403
If-Modified-Since header, 96, 533
<IFRAME> tag, 329-330, 477
attributes
align, 330
frameborder, 329
height, 330
marginheight, 329
marginwidth, 329
name, 329
scrolling, 330
src, 329
width, 330
listing, 330
IIS (Internet Information Server), 773-774
disadvantages, 774
technical support, 774
<ILAYER> tag, 349
attributes
above, 350
background, 350
below, 350
bgcolor, 350
clip, 350
left, 351
name, 351
top, 351
visibility, 351
width, 351
zindex, 351
Image Composer, 718, 741
features, 718
sprites, 718
system requirements, 718
Web site, 718
image control, 606
image maps, 847
Backstage Designer, 747
client-side, 241, 264, 268-271
advantages, 268
<AREA> tag, 269
browser support, 271
designating, 242
enabling, 269
<MAP> tag, 269-271
editors, 271-274
LiveImage, 273-274
Map This!, 272-273
MapEdit, 271-272
<FIG> tag proposal, 270
graphical clues, 241
hotspots, 264
Macromedia ImageMap PowerApplet, 585-586
navigation, 475
disadvantages, 475
server-side, 241, 264-268
designating, 241
including, 265
MAP files (creating), 266-268
script file path, 265
Travelution Web site, 494
see also <MAP> tag
images
aligning, 235-238
bottom alignment, 237
left alignment, 237
middle alignment, 236
right alignment, 238
top alignment, 235
background, 242-243, 950
BACKGROUND attribute, 149
placement, 950
repeating, 950
Backstage Designer, 747
borders, 240
buffers, 240
download time, 306
floating, clear attribute, 161
fractals, 727
Fractint, 727
Winfract, 727
Xfractint, 728
GIFs, animated, 668
identifying type, 235
<IMG> tag, 234-235, 847
inline, clear attribute, 161
interface style, 448-449
linked, navigation, 475
list markers, 957
LOW SRC, 704
multimedia, 247
plug-ins, 674-676
sizing, 239
spacer, layout, 460-461
substitute content, 243-244
text descriptions, 235
see also graphics
<IMG> tag, 84, 234-235, 847, 869-871
attributes
align, 235-238
alt, 235
border, 240

IPR2024-00143
Apple EX1011 Page 519

dynsrc, 256-257
height, 239
hspace, 240
ismap, 241
src, 235
usemap, 242, 269
vspace, 240
width, 239
W3C HTML 3.2 specification, 920-923
implementing JavaScript objects, 549
implicit labels, 332
#IMPLIED keyword, 64
Import Web wizard (Explorer), 733
in-flow layers, 349
creating, 358
InContext WebAnalyzer, 701-702
display, 702
reports, 702
setting options, 702
Web site, 701
indentation (text), 952
indenting
<BLOCKQUOTE> tag, 166
text, 952
informational response codes, 526
Informix, 687
Web site, 687
inheritance, 847
Java, 576
init method, 576
initializing Java applets, 576
inline images
clear attribute, 161
W3C HTML 3.2 specification, 920-923
inline keyword, display attribute, 956
Inline node, 638
inline style sheets, 366-368, 947
input fields, type attributes, 915-917
<INPUT> tag, 278-283, 596-598, 874-875
input types
CHECKBOX, 281
HIDDEN, 281
IMAGE, 283
PASSWORD, 280
RADIO, 282
RESET, 283
SUBMIT, 282-283
TEXT, 279-280
W3C HTML 3.2 specification, 915-918
type attributes, 915-917
inputform.alx (listing), 622, 625-626
inserting, HTML Layout control, 621-622
installing Personal Web Server, 739
instantiation, 549
InStr( ) function, 619
integrated services digital network (ISDN), 847
integrating plug-ins, 654
intellectual property, 847
interfaces
controls, 442-446
hardware elements, 442
selecting, 445-446
software elements, 443
defined, 847
designing, 436-439
audience, 440-441
color, 446-448
content, 441
criteria, 437
CyberMad, 499-503
image styles, 448-449
look and feel of site, 441-442
media considerations, 438-439
Microsoft Sitebuilder site, 437
site goals, 439-440
themes, 439
Travelution, 488-492
HTML as, 829-830
nonlinear information systems, 10
NLS, 10
internal multimedia, 246-249
ActiveX controls, 249
applets, 247-248
images, 247
plug-ins, 248
International Organization for Standardization (ISO), 39, 795, 847
internationalization, 335
character encoding, 794-799
7-bit ASCII, 795-796
8-bit encodings, 796-797
16-bit encodings, 797-798
bit combinations, 795
versus character sets, 795
ISO 10646, 798-799
fonts, 810-811
Cyberbit, 810
glyph servers, 810
OpenType, 810
forms, 803-804
accept-charset attribute, 803-804
language identification, 806
extended identifiers, 806
lang attribute, 806
language-specific markup, 807-809
alignment, 809
cursive joining behavior, 808
hyphenation, 809
quotation marks, 808-809
writing direction, 807-808
MIME, 799-800
charsets, 799
specification, 799
RFC 2070 document, 794
HTML 4, 322

Internet
access
dialup, 843
gophers, 845
defined, 847
TCP/IP, history, 89-90
Internet Explorer, 32, 125-128, 847
4.0 version, 125
ActiveX controls, 260-262
align attribute, 261
codebase attribute, 261
codetype attribute, 261
including, 261
background sounds, 257
bullets, 190
extensions, 128
FrontPad, 125
HTML extensions, 65
multilingual support, 669-670
plug-ins, 259-260
disadvantages, 259
embedding, 260
Shockwave, 259
scrolling marquees, 257-258
aligning, 258
applets, 258
background color, 258
behaviors, 257
buffers, 258
delay (setting), 258
direction, 258
formatting text, 257
loops, 258
size, 258
versions, 32
versus Navigator, 118-121
video, dynsrc attribute, 256-257
see also pocket Internet Explorer
Internet Explorer Web site, 963
Internet nodes, URIs, 103
Internet Protocol (IP), 848
Internet service provider (ISP), 848
intranets, 848
intrinsic events, 328
onBlur, 328
onChange, 328
onClick, 328
onFocus, 328
onLoad, 328
onMouseOut, 328
onMouseOver, 328
onSelect, 328
onSubmit, 328
onUnload, 328
intrinsic HTML controls, 603
attaching scripts to, 596-598
IP (Internet Protocol), 848
IP addresses, 848
ISDN (integrated services digital network), 847
<ISINDEX> tag, 77, 144-145, 863
prompt attribute, 144
W3C HTML 3.2 specification, 896
ismap attribute, <IMG> tag, 241
ISO (International Organization for Standardization), 39, 795, 847
ISO 646, see 7-bit ASCII, 795
ISO 8859 standards, 796
ISO 10646, 798-799
BMP (Basic Multilingual Plane), 799
ISO Latin-1, character entities (code listing), 940-942
ISP (Internet service provider), 848
versus purchasing Web servers, 767-768
italics, 948
<CAPTION> tag, 204
definition lists, 193
<I> tag, 171
term definition, 167
iWORLD Web site, 772

J

J-Track applet, 587
JARS (Java Applet Rating Service), 588
JASC, Inc. Web site, 715
Jasmine database, 686
Web site, 686
Java, 568, 848
advantages, 568
application sizes, 569
error handling, 570
object-oriented code, 569
platform-independence, 568-569
reliability, 570
applets, 571-572
Animator, 582-584
Applet Viewer, 572-573
Calendar, 584
class declaration, 574-575
creating, 574
destroying, 576
displaying, 575-577
example of, 577-578
HTML markup, 578-579
initializing, 576
J-Track, 587
Macromedia ImageMap PowerApplet, 585-586
online sources, 588-589
overriding, 576
starting, 576
stopping, 576
Ticker, 586-587
applications, 571-572
browser compatibility, 571
compiler, 570
inheritance, 576
JavaScript, compared, 544, 572
navigation, 477
scripts, 579-582
security, 570
byte code verification, 570
file access, 570
memory layout, 570

IPR2024-00143
Apple EX1011 Page 520

Java Animator, 250-255
directory structure, 251
graphic files, 251
implementing animation, 252-253
location, 250
parameters, 254-255
sound format, 252
Java Applet Rating Service (JARS), 588
JavaScript, 544-545, 848
advantages, 545
API, 546
browsers
compatibility, 546-547
controlling, 561
case sensitivity, 547
functions, 550-551
HTML markup
<SCRIPT> tag, 553-555
comment tags, 555
versus Java, 544, 572
layers, 358-361
listing, 359-360
methods, 551-553
navigation, 477
objects, 547-549
classes, 549
defining, 547
implementing, 549
properties, 550
online documentation, 546
online resources
Beginner's Guide to JavaScript, 562
Gamelan, 562
Netscape, 562
newsgroups, 564
Voodoo JavaScript Tutorial, 563
sample applications
form validation, 557-559
random numbers, 559-560
status bar messages, 560-561
scripts
embedding, 555-556
hiding, 555
style sheets, 400-401, 848
backgroundImage property, 409
block-level element properties, 407-408
browser support, 401
class attribute, 403
classes, 401
color property, 409
display property, 409
font properties, 404-405
ids attribute, 403
<LINK> tag, 402
listStyleType property, 409
precedence order, 410
<SPAN> tag, 403
style attribute, 403
<STYLE> tag, 402
text properties, 405-406
whiteSpace property, 409
JavaSoft Web site, 255, 588
JDK, downloading, 250
JPEG (Joint Photographic Experts Group Format), 848
justification, 376, 952
see also alignment

K

Kai's Power Tools (Photoshop filters), 714
<KBD> tag, 83, 169, 867
Kbps (kilobits per second), 849
kerning, 849
keywords
ATTLIST, 62
BASESET, 42
CDATA, 62
CHARSET, 42
content model, 59
ANY, 59
CDATA, 59
EMPTY, 59
RCDATA, 59
DEF (VRML), 640
DESCSET, 42
#FIXED, 64
ID, 63
#IMPLIED, 64
NAME, 62
NMTOKEN, 62
NUMBER, 63
#PCDATA, 61
PUBLIC, 786
#REQUIRED, 64
SYSTEM, 786
Kid's News style sheet, 395-397
kilobits per second, 849

L

label control, 607
properties, 608-609
<LABEL> tag, 331-332
attributes
accesskey, 332
class, 331
dir, 332
disabled, 332
for, 331
id, 331
lang, 332
onBlur, 332
onClick, 332
onFocus, 332
style, 331
title, 332
labels (forms), 331-332
explicit, 332
implicit, 332
lang attribute, 806
<BUTTON> tag, 332
<FIELDSET> tag, 333
<LABEL> tag, 332
language attribute, <SCRIPT> tag, 327, 593

language identification, 806
extended identifiers, 806
lang attribute, 806
laptops, 116
connections, 116
monitors, 116
Last-Modified header, 100, 533, 536
<LAYER> tag, 349, 416
attributes
above, 350
background, 350
below, 350
bgcolor, 350
clip, 350
left, 351
name, 351
top, 351
visibility, 351
width, 351
zindex, 351
methods
moveAbove, 353
moveBelow, 354
moveTo, 354
offset, 353
resize, 354
layers, 348-349, 849
clipping rectangles, 357
listing, 357
considerations, 348-349
browser support, 348
download time, 349
W3C standards, 349
creating, 354-357
stacked layers page listing, 356
three-layer page listing, 355
in-flow, 349
creating, 358
JavaScript, 358-361
listing, 359-360
language support, 354
<LAYER> and <ILAYER> tag attributes, 350-351
methods, 350
moveAbove, 353
moveBelow, 354
moveTo, 354
offset, 353
resize, 354
out-flow, 349
properties, 351-353
versus frames, 348
visibility, 348
layout, 292-301, 452
animated graphics, 455
backgrounds, 456
classic layout example (columns), 296-297
conventional layout example, 293-295
CSS, 454-455
CyberMad Web site, 503-507
tables, 504-505
defined, 849
elements, 453
fonts, 454
formal page example, 299-301
frames, 344-347
nesting frames, 346
simple layout, 344-346
graphics, 455, 465-466
file sizes, 465
graphics-only programs, 465-466
horizontal lines, 176
links, 457
lists, 311-312
outer list listing, 311-312
modern layout example, 295
page dimensions, 457-458
plug-ins, 458
programmed controls, 457
space, 293
spacer images, 460-461
vertical control, 461
style attributes, 376-378
margin-left, 376
margin-right, 376
margin-top, 376
text-align, 376
text-indent, 376
tables, 456-457, 461-465
absolute widths, 456
guidelines, 462-463
nested tables, 464-465
newspaper columns, 463-464
relative widths, 456, 464
technical information page example, 297-298
templates, 301-308
banners, 302-303
content area, 303
creating home pages from, 308-311
creating pages from, 306-308
structure listing, 301
template.htm file listing, 305
text, 453-454
Travelution Web site, 492-494
animated GIFs, 493
flaw, 493
fonts, 492
graphics, 492
tables, 492
without tables, 458-461
LCD (liquid crystal display), 849
leading, 849
left alignment, 156
left attribute, <LAYER> and <ILAYER> tags, 351
left value, clear attribute, 161
legal issues
copyrights, 842
intellectual property, 847
watermarks, 859
length (fonts), 949
letter-spacing attribute, 951
<LI> tag, 181-182, 866-867
lighting (VRML), 637

IPR2024-00143
Apple EX1011 Page 521

line breaks
<BLOCKQUOTE> tag, 166

 tag, 159-161
line height (text), 953
line spacing, <BR> tag versus <P> tag, 160
line-height attribute, 374, 953
linear information, 849
versus nonlinear, 4
lineHeight property, JavaScript style sheets, 405
lines
horizontal, <HR> tag, 176
table cells, 203
LINK attribute, <BODY> tag, 151
Link header, 533
Link Wizard, 745
<LINK> tag, 77, 145-146, 402, 863-864, 947
attributes
href, 145, 325
media, 325
name, 145
rel, 145, 325
rev, 145, 325
title, 145, 325
type, 325
CSS, 325
listing, 402
W3C HTML 3.2 specification, 898-899
linking
style sheets, 366, 369, 390, 947
format, 947
syntax, 366
superscripts to endnote, 173
links, 222
<ADDRESS> tag, 165
<BASE> tag, 143-144
HREF attribute, 143
broken
fixing, 745
WebAnalyzer utility, 701-702
Xpire Plus utility, 703-704
color, 224
active links, 151
LINK attribute, 151
visited links, 152
connecting within documents, 230
referencing filenames, 231
creating, 223-224
databases (Access 97), 682
defining type, 227
e-mail addresses, 226-227
hotspots, 846
HTML 4, 322
images, navigation, 475
maintaining
Backstage Manager, 744-745
NetObjects Fusion, 754
navigation, 474
disadvantages, 474
page layout, 457
relative, 224-225
<BASE> tag, 225-226
underlining (<U> tag), 171
viewing, Explorer, 731
XML, 788-789
Linux, 849
liquid crystal display (LCD), 849
List view (Script Wizard), 610
list-style attribute, 957
list-style position attribute, 957
list-style-image attribute, 957
list-style-type attribute, 956
listbox control, 607
<LISTING> tag, 163
listings
3×3 table, 209

 tag, 160
CGI script table example, 215-216
classic layout example with columns, 296-297
clipping rectangles, 357
column spanning, 209-210
conventional Web page layout example, 294
drag-and-drop graphics, 421-422
dynamic ActiveX controls, 612-613
dynamic record presentation, 424-426
dynamic row sorting, 428-429
dynamic text, 418
embedding VRML billiard table in HTML, 644-645
formal Web page example, 299-300
frame layout code, 341
guest book form example, 288
<HEAD> tag, 141
horizontal rules, extensions versus standards, 112
HTML document to display MyApplet, 578
HTML document without structural form, 138
<HTML> tag, 140
in-flow layers page, 358
inline frame page, 330
inputform.alx, 622, 625-626
internal anchors
connecting links, 230
definition term in a separate document, 231
enclosing destination headings, 229
placing tags in front of headings, 229
referencing filenames, 231
referencing glossary page terms, 231
JavaScript definition of a squirrel in an HTML document, 549
JavaScript form validation script, 558
JavaScript HTML document, 554

JavaScript methods, 552-553
JavaScript object definition for a squirrel, 548
JavaScript script in HEAD portion, 556
layers page with JavaScript, 359-360
layout example with alignment, 295
<LINK> tag, 402
Missoula Rural Fire District site
guest book, 313-315
home page created from a template, 308-310
list page layout, 311-312
page created from template.htm template, 306-307
template banner, 302
template content area, 303-304
template structure, 301
template.htm file, 305
nested frames, 347
ordered lists
interrupting, 184
list item tag, 182
nested lists, 187
<OL> tag, 180
start attribute, 185
with <BR>, <P> tags, 183
<P> tag, 159
page with three layers, 355
paragraph formatting, 158
<PRE> tag, 162
reading and decoding form data in Perl 5, 523-524
registration form, 516-517
registration form encoded data, 517
row spanning, 211
samplelayout.htm, 627
script attached a button's click event, 596
script written inline, 597
simple applet, 577
simple frame layout (three frames), 345
simple frame layout (two frames), 344
<SPAN> tag, 403
stacked layers page, 356
<STYLE> tag, 402
table with columns and hanging headline, 217-218
technical information Web page example, 298
<TITLE> tag, 142
unordered lists, 189
bullets, 191
VRML billiard table, 638-639
VRML frame HTML definition document, 645
wblteam.txt file, 430
Web page with frames, 344
Web page with frames, Left frame code, 344
lists, 311-312
compact attribute, 192
definitions, 167
formatting, 957
glossary, <DL> tag, 192-194
HTML 3.2 W3C specification, 903-905
image markers, 957
JavaScript style sheet, 409
logical styles, 194
marker types, 956
ordered
<LI> tag, 181
nested lists, 187-189
<OL> tag, 180-189
outer list listing, 311-312
unordered
nested, 191
type attribute, 190-192
<UL> tag, 189-190
Lists element, 82
listserv, 849
listStyleType property, JavaScript style sheets, 409
LiveImage editor, 273-274
LiveImage Web site, 274
loading Web pages, audio, 457
local projects (Backstage Manager), 744
Location header, 99, 533
logical list styles, 194
logical markup, 849
elements, 83
CITE, 83
CODE, 83
DFN, 83
EM, 83
KBD, 83
Q, 83
SAMP, 83
STRONG, 83
VAR, 83
logical styles, 163-164
<ADDRESS> tag, ordered lists, 184
review, 170
logical tags, 164
loop attribute, <MARQUEE> tag, 258
loose casting, 550
Lotus Approach 97, 681-682
Web site, 681
LOW SRC images, 704
Lycos Web site, 479
Lynx, 130, 850
Web site, 963

M

M. Casco Web site, 833
Macintosh
animated GIF files (creating), 256
PageMill editor, 693
MacOS, Web servers, 770
Macromedia
Director, 725
system requirements, 725
Web site, 725

IPR2024-00143
Apple EX1011 Page 522

Flash, 725
system requirements, 725
Web site, 725
FreeHand, 721-722
features, 722
Shockwave plug-in, 722
system requirements, 722
Web site, 721
ImageMap PowerApplet, 585-586
Shockwave, 665-666
Web site, 666
xRes, 718
Macromedia Backstage, see Backstage
mailing lists
Apache, 773
listserv, 849
Photoshop, 715
mailto scheme, 107
mailto: protocol, 226-227
maintenance, site maps, 481
Manager (Backstage), 742-745
maintaining links, 744-745
drag-and-drop feature, 745
Link Wizard, 745
projects, elements, 743-744
remote administration, 743
MAP files, creating
CERN servers, 266-267
NCSA servers, 267-268
Map This! editor, 272-273
Web site, 273
<MAP> tag, 84, 269-271, 873
W3C HTML 3.2 specification, 926-927
MapEdit editor, 271-272
Web site, 272
margin attribute, 953
margin-bottom attribute, 953
margin-left attribute, 376, 953
margin-right attribute, 376, 953
margin-top attribute, 376, 953
marginheight attribute
<FRAME> tag, 342
<IFRAME> tag, 329
margins
attributes, 953
floating images, 161
JavaScript style sheet properties, 407
setting, 376-377
marginwidth attribute
<FRAME> tag, 342
<IFRAME> tag, 329
markup, 850
defined, 23
descriptive, 36-39
language-specific, 807-809
alignment, 809
cursive joining behavior, 808
hyphenation, 809
quotation marks, 808-809
writing direction, 807-808
logical, 849
physical, 853
procedural, 36-39
proprietary, 70-71
see also elements
<MARQUEE> tag, 257-258
attributes
align, 258
behavior, 257
bgcolor, 258
direction, 258
height, 258
hspace, 258
loop, 258
scrollamount, 258
scrolldelay, 258
vspace, 258
width, 258
marquees, 257-258
formatting text, 257
scrolling aligning, 258
scrolling applets, 258
scrolling background color, 258
scrolling behaviors, 257
scrolling buffers, 258
scrolling delay (setting), 258
scrolling direction, 258
scrolling loops, 258
scrolling size, 258
maxlength attribute, text fields, 280
Mbps (million bits per second), 850
measuring systems, tables, 201
media attribute
<LINK> tag, 325
<STYLE> tag, 326
memex, 8-9, 109
memory
allocating (Java), 570
RAM, 854
<MENU> tag, 194, 866
W3C HTML 3.2 specification, 906
message boxes, displaying, 601
<META> tag, 78-79, 146-148, 864
content attribute, 78
expiration dates, 78
hppt-equiv attribute, 78, 146
variables, 146-147
name attribute, 146-147
refreshing, 79
W3C HTML 3.2 specification, 897
method attribute, <FORM> tag, 276
Method request header, 529-530
methods
Alert, 601
destroy, 576
drawString, 577
GET, 94, 276, 516
encoded data listing, 517
registration form listing, 516-517

request header, 530-531
request headers, 529
HEAD, 94
init, 576
JavaScript, 551-553
layers, 350
moveAbove, 353
moveBelow, 354
moveTo, 354
offset, 353
resize, 354
paint, 575-577
pasteHTML( ), 420
POST, 94, 276-277
rangeFromElement( ), 420
start, 576
stop, 576
Write, 601
see also functions
micro minis, see HPCs
Microsoft FrontPad site, 965
Microsoft FrontPage, see FrontPage
Microsoft HTML Layout control, see HTML Layout control
Microsoft Internet Explorer, see Internet Explorer
Microsoft Site Builder Network Web site, 960
Microsoft Sitebuilder Web site, 437
MIDI (Musical Instrument Digital Interface), 851
million bits per second (Mbps), 850
MIME (Multipurpose Internet Mail Extensions), 851
HTML 4 support, 335
internationalization, 799-800
charsets, 799
specification, 799
MIME-Version header, 533
minimization indicators, 58
MINIMIZE class, HTML 4.0 declaration, 54-55
modems
cable-modem, 841
defined, 850
monitors
laptops/notebooks, 116
NCs, 116
PCs, 115
monospace fonts, 371, 850, 948
<CODE> tag, 168
<KBD> tag, 169
<PRE> tag, 162
<TT> tag, 172
Moore, Gordon (Moore's Law), 850
Mosaic, 28-29, 118, 130
bullets, 190
Motion Picture Experts Group (MPEG), 850
moveAbove method, 353
moveBelow method, 354
moveTo method, 354
Mozilla, 29-31, 850
HTML extensions, 30-31
problems, 31
MPEG (Motion Picture Experts Group), 850
Missoula Rural Fire District Web site, 292
guest book, 313-315
listing, 313-315
home page, 308-311
nested list page, 311-312
template, 301-308
multicolumn directory lists, <DIR> tag, 194
MultiGIF, 725
multilingual support, plug-ins, 669-672
multimedia
ActiveX controls, 260-262
align attribute, 261
class IDs, 261
codebase attribute, 261
codetype attribute, 261
including, 261
animated GIF files, 255-256
creating, 256
download time, 255
background audio, 257
defined, 246, 850
external, 246, 249-250
connecting to, 250
helper applications, 249
RealAudio, 249
HTML level 3.2, 250
hypermedia, 846
internal, 246-249
ActiveX controls, 249
applets, 247-248
images, 247
plug-ins, 248
Java Animator, 250-255
directory structure, 251
graphic files, 251
implementing animation, 252-253
location, 250
parameters, 254-255
sound format, 252
Macromedia Director, 725
system requirements, 725
plug-ins, 259-260, 663-665, 668
Cosmo Player, 667
disadvantages, 259
embedding, 260
Shockwave, 259, 665-666
Surround Video, 666
WIRL, 667
pure HTML, 822-823
scrolling marquees, 257-258
aligning, 258
applets, 258
background color, 258
behaviors, 257
buffers, 258
delay (setting), 258
direction, 258
formatting text, 257
loops, 258
size, 258
video, Internet Explorer, 256-257

IPR2024-00143
Apple EX1011 Page 523

Multipurpose Internet Mail Extensions, see MIME
Musical Instrument Digital Interface (MIDI), 851
MUX protocol, 102

N

name attribute
<A> tag, 228
<FRAME> tag, 342
<IFRAME> tag, 329
<LAYER> and <ILAYER> tags, 351
<LINK> tag, 145
<META> tag, 146-147
<OBJECT> tag, 334
<PARAM> tag, 335
NAME keyword, 62
name/value pairs, 518
ampersands (&), 518
name field, 518
passing to servers, 518
splitting, 522
value field, 518
naming rules, HTML 4.0 declaration, 53
National Center for Supercomputing Applications (NCSA), 28, 851
Navigate with an Accent plug-in, 670
navigation, 473-478, 851
ActiveX controls, 477
CSS, 475
CyberMad Web site, 507-508
links, 508
navigation menu, 507
Table of Contents page, 508
designing system, 470-472
clarity, 472
outlining site, 471
placement, 472
frames, 476-477
advantages, 476
disadvantages, 476
image links, 475
advantages, 475
disadvantages, 475
image maps, 475
disadvantages, 475
Java, 477
JavaScript, 477
links, 474
disadvantages, 474
search engines, 481-483
adding, 482
advantages, 481
customizing output, 482-483
parameters, 483
Web server support, 482
site maps, 478-481
color-coding, 481
Indian government example, 480
Lycos Web site, 479
maintenance, 481
text-only, 478
Travelution Web site, 494-496
image map, 494
site map, 494
visual cues, 477-478
Navigator, 122-124
ActiveX controls, 260-262
align attribute, 261
codebase attribute, 261
codetype attribute, 261
including, 261
extensions, 123-124
multilingual support, 669-670
plug-ins, 259-260
disadvantages, 259
embedding, 260
Shockwave, 259
versions, 31
versus Internet Explorer, 118-121
see also Netscape
Navigator Gold, 696-697
features, 697
posting pages, 697
NCs (network computers), 116, 851
monitors, 116
NCSA (National Center for Supercomputing Applications), 28, 266, 774, 851
bugs, 774
downloading, 774
NCSA Mosaic Web site, 964
NCSA servers, MAP files, 267-268
circle keyword, 268
default keyword, 268
point keyword, 268
poly keyword, 268
rect keyword, 268
Nelson, Ted, 10-11
Xanadu, 11
nested frames, 346-347
listing, 347
nested lists, 187-189
unordered, 191
nested tables, 464-465
background colors, 464
browser support, 464
CyberMad Web site, 505
Travelution Web site, 492
nesting Frame objects, 605
Net.Data, 685
security, 685
Web macro, 685
Web site, 685
Netcraft Web site, 771
NetImpact Dynamo, 683
gateways supported, 683
HTML authoring tools, 683
Web site, 683
NetMeeting, 674
NetObjects Fusion, 730, 751-756
Assets view, 754
components, 752
Page view, 753
Publish view, 755-756

Site view, 752
Style view, 754
system requirements, 752
Netra j Web server, 776
Netscape, 29, 851
color palette, 447
Communicator 4.0, 122
collaborative browsing, 122
HTML extensions, 30-31, 65, 416-417
problems, 31
layers, 849
Web site, 414, 562, 696
see also Navigator
Netscape Composer, see Composer
Netscape DevEdge OnLine Web site, 960
Netscape Web servers, 774-775
cost, 774
evaluation copies, 774
technical support, 775
Network Computers (NCs), 851
news scheme, 107
newsgroups
alt.lang.java, 589
HTML resources, 966
Photoshop, 715
newsletters, style sheets, 388-390
newspaper columns, 463-464
<NEXTID> tag, 864
NLS (oN-Line System), 10
NMTOKEN keyword, 62
NNTP Web servers, 763
<NOBR> tag, 868
nodes
hypertext, 222
VRML, 635
Appearance, 641
computers, 851
fields, 635
geometry, 636
Group, 638
Inline, 638
lighting, 637
PositionInterpolator, 637
Shape, geometry field, 641
SphereSensor, 637
Timer, 637
TouchSensor, 637, 643
Transform, 638, 642-643
Viewpoint, 637, 643
Web pages, 851
<NOEMBED> tag, 873
<NOFRAME> tag, 882
none keyword, display attribute, 956
nonlinear information systems, 4-6, 851
history, 6-10
Bush, Vannevar, 8-9
Engelbart, Doug, 10
memex, 8-9
NLS, 10
notched cards, 7-8
reading machines, 6-7
user interfaces, 10
HyperCard, 12-13
hypertext, 10-16
Atkinson, Bill, 11-13
dividing content, 15
Nelson, Ted, 10-11
providing author context, 16
providing reader context, 15-16
versus linear, 4
WWW, 13-14
Berners-Lee, Tim, 13-14
Xanadu, 11
noresize attribute, <FRAME> tag, 342
normal fonts
font-style attribute, 948
font-variant attribute, 948
font-weight attribute, 948
Normal Web template (Explorer), 733
NOSCRIPT element, 80
noshade attribute, <HR> tag, 177
notebooks, 116
connections, 116
monitors, 116
Novartis Web site, 440
Now( ) function, 604
nowrap attribute, <TD> and <TH> tags, 212
NUMBER keyword, 63
number signs (#), URL encoding, 521
numbered lists, <OL> tag, 180-184
start attribute, 184-186
type attribute, 187-189
numbering ordered lists, 186

O

object model (Dynamic HTML), 414-415
object-oriented programming, 569, 852
<OBJECT> tag, 84, 333-334, 871
attributes
align, 334
alt, 334
border, 334
CLASSID, 248
classid, 333
codebase, 333
CODETYPE, 248
codetype, 334
data, 333
declare, 333
height, 334
hspace, 334
id, 333
name, 334
shapes, 334
standby, 334
title, 334
type, 333
usemap, 334

IPR2024-00143
Apple EX1011 Page 524

vspace, 334
width, 334
ActiveX controls, 249, 261
applets, 248
level 4 expansion, 247
plug-ins, 248
objects
Backstage, 742, 750
defined, 852
embedded, 843
HTML object model, 598
Document, 601
Elements, 603-604
Form, 601-603
hierarchy, 599-600
variables, 604
Window, 601
identifying, DEF keyword (VRML), 640
inserting, 333
OBJECT element, 333-334
PARAM element, 334-335
JavaScript, 547-549
classes, 549
defining, 547
implementing, 549
properties, 550
oblique fonts, 948
ODBC (Open Database Connectivity), 852
offset method, 353
O'Reilly and Associates Web site, 775
<OL> tag, 180-184, 866
attributes
start, 184-186
type, 187-189
W3C HTML 3.2 specification, 904-905
oN-Line System (NLS), 10
onBlur attribute, <LABEL> tag, 332
onBlur event, 328
onChange event, 328
onClick attribute, <LABEL> tag, 332
onClick event, 328
onFocus attribute, <LABEL> tag, 332
onFocus event, 328
online books, ordered lists, 187
onLoad event, 328
onmousemove event, 423
onMouseOut event, 328
onMouseOver event, 328
onSelect event, 328
onSubmit event, 328
onUnload event, 328
Open Database Connectivity, see ODBC
open protocols, 91
OpenIngres/Internet Commerce Enabled database, see ICE database
OpenType fonts, 810
Web site, 810
Opera Web site, 964
operating system, see OS
<OPTION> tag, 875
optionbutton control, 607
Oracle, 684-685
Developer/2000 report writing module, 684
HTML forms, 684
HTML reports, 684
ordered lists
HTML 3.2 W3C specification, 904-905
<LI> tag, 181
nested lists, 187-189
numbering, 186
<OL> tag, 180-184
start attribute, 184-186
type attribute, 187-189
Orig-URL header, 533
OS (operating system), 852
HPCs, 117
Windows, 859
OS/2, Web servers, 771
OTHER class, HTML 4.0 declaration, 55
out-flow layers, 349
out-of-line links, 789
overriding Java applets, 576

P

<P> tag, 82, 158-159, 865
versus <ADDRESS> tag, 165
versus <BR> tag, 160
with <CODE> tag, 168
versus <DIV> tag, 163
ordered lists, 181-182
W3C HTML 3.2 specification, 903
packets, demultiplexing, 91
padding
JavaScript style sheet properties, 407
text boxes, 953-954
padding attribute, 954
padding-bottom attribute, 953
padding-left attribute, 953
padding-right attribute, 953
padding-top attribute, 953
page layout, 292-301
classic layout example (columns), 296-297
conventional layout example, 293-295
formal page example, 299-301
frames, 344-347
nesting frames, 346
simple layout, 344-346
lists, 311-312
outer list listing, 311-312
modern layout example, 295
space, 293
style attributes, 376-378
margin-left, 376
margin-right, 376
margin-top, 376

text-align, 376
text-indent, 376
technical information page example, 297-298
templates, 301-308
banners, 302-303
content area, 303
creating home pages from, 308-311
creating pages from, 306-308
structure listing, 301
template.htm file listing, 305
Page view (NetObjects Fusion), 753
PageDraw, 756
PageMill, 692-694
features, 693
hidden HTML option, 693
Macintosh version, 693
Web site, 966
pages
content, 842
data-aware, 423-430
binding to records, 424-428
data source control, 424
dynamic presentation listing, 424-426
features, 424
generating/sorting table rows, 428-430
defined, 852
dynamic, 843
hits, 846
nodes, 851
paint method, 575-577
paint programs, 712
PaintShop Pro, 715-716
download site, 715
features, 715
strengths, 715
system requirements, 716
weaknesses, 715
Palmtops, see HPCs
paragraphs
<BLOCKQUOTE> tag, 166
formatting, 158
HTML 3.2 W3C specification, 903
level 0 HTML, 22
ordered lists, 183
<PARAM> tag, 334-335, 579, 872
attributes
name, 335
value, 335
valuetype, 335
parameter entities, 56
case sensitivity, 56
declaring, 56
percent signs (%), 56
versus general, 57-58
parameters
charset, 802
defined, 852
Java Animator, 254-255
passing to applets, 579
search engines, 483
values, 858
parsing, 852
party invitations, style sheets, 394-395
Password control, 603
password fields, 280
passwords, 852
pasteHTML( ) method, 420
PATH environment variable, 539
PATH INFO environment variable, 515, 539
path information (CGI programs), 518-519
ACTION field, 518
PATH TRANSLATED environment variable, 539
paths, 852
see also URLs
***PC Magazine Online* Web site, 706**
PC Resources for Photoshop Web site, 714
#PCDATA keyword, 61
PCs, 115
connections, 115
monitors, 115
PDA (personal digital assistant), 852
see also HPCs
percent signs (%), 56
horizontal lines, 177
table alignment, 201
URL encoding, 519-521
percentage value (fonts), 949
Perl (Practical Extraction and Report Language), 853
permissions, FrontPage
administrator, 734
author, 734
end user, 733
screening, 734
personal digital assistant (PDA), 852
Personal Home Page Wizard, 737
Personal Web Server, 739
download site, 739
installing, 739
managing, 739
number of users, 739
Personal Web template (Explorer), 733
Pesce, Mark, 633
Philips/Magnavox Web site, 132
Photoshop Paradise Web site, 715
phrase elements, W3C HTML 3.2 specification, 914
physical markup, 853
elements, 83-84
B, 84
BIG, 84
I, 84
SMALL, 84
STRIKE, 84
SUB, 84
SUP, 84

IPR2024-00143
Apple EX1011 Page 525

TT, 84
U, 84
physical styles, 170
<B> tag, 171
<FONT> tag, 173
color attribute, 175
size attribute, 173-175
<I> tag, 171
ordered lists, 184
<STRIKE> tag, 171-172
<SUB> tag, 172-173
<SUP> tag, 172-173
<TT> tag, 172
<U> tag, 171
PICS (Platform for Internet Content Selection), 853
pipelining, 101
pixel sizes, versus display sizes, 113
pixels, 853
horizontal lines, 177
table width, 199
<PLAINTEXT> tag, 163
Platform for Internet Content Selection (PICS), 853
platform independence, 70-72, 853
proprietary markup, 70-71
pure HTML tags, 816
platforms, 853
HPCs, 116-117
connections, 117
operating systems, 117
idiosyncrasies, pixel versus display size, 113
laptops/notebooks, 116
connections, 116
monitors, 116
NCs, 116
monitors, 116
PCs, 115
connections, 115
monitors, 115
play.dtd, 787-788
play.xml, 786
DOCTYPE statement, 786-787
XML declaration, 786
plug-ins, 259-260, 652-654, 853
animation, 663-665, 668
Cosmo Player, 667
Shockwave, 665-666
Surround Video, 666
WIRL, 667
audio, 659-663
Backstage Designer, 748
business, 676-677
communications/conferencing, 672-674
disadvantages, 259
embedding, 260
external, 668-669
filters, Adobe Photoshop, 714
image, 674-676
integrating, 654
layout, 458
multilingual support, 669-672
multimedia, 248
Shockwave, 259, 722, 855
video, 654-659
plus signs (+), 60
PNG (Portable Network Graphic), 853
Pocket Internet Explorer, 130-131
ports, 853
PositionInterpolator node, 637
POST method (HTTP), 94, 276-277
POV-Ray, 727
system requirements, 727
Web site, 727
Power Translator 6.0, 671
PowWow plug-in, 672-673
Practical Extraction and Report Language, see Perl
Pragma header, 98, 533
<PRE> tag, 82, 161-162, 868
W3C HTML 3.2 specification, 906-907
preformatted text
HTML 3.2 W3C specification, 906-907
<LISTING> tag, 163
<PLAINTEXT> tag, 163
<PRE> tag, 161-162
<XMP> tag, 163
price lists, style sheets, 390-393
procedural markup, 36-39
program code
<CODE> tag, 167-168
<KBD> tag, 169
<SAMP> tag, 169
programming, object-oriented, 569
programming languages, HTML as, 833
M. Casco Web site, 833
Progressive Network's EasyStart Server, 657
Project Web template (Explorer), 733
projects (Backstage Manager), 743-744
database queries, 743
discussion groups, 743
local, 744
remote, 744
user profiles, 743
prompt attribute, <ISINDEX> tag, 144
proofreader marks, underlining (<U> tag), 171
properties
ActiveX controls, 608-610
JavaScript objects, 550
JavaScript style sheets
align, 408
backgroundImage, 409
block-level elements, 407-408
clear, 408
color, 409
display, 409
fonts, 404-405
height, 408

listStyleType, 409
text, 405-406
whiteSpace, 409
width, 408
layers, 351-353
Property or Method Not Supported by this Object message, 599
proportional fonts, 853
proprietary markup, 70-71
protocols
connectionless, 90
defined, 103, 854
ftp, 89, 844
HTTP, 88, 91-92
connections, 91-93
entities, 99-101
requests, 91-93
responses, 91, 96-99
transactions supported, 91
mailto:, 226-227
mixing, HTTP-NG connections, 102
MUX, 102
open, 91
SHTTP, 855
stateless, 91
TCP, 857
TCP/IP, 90-91
TCP, 91
see also schemes
proxy gateways, 854
proxy GET method request header, 531
proxy servers, 854
Public header, 533
public identifiers, 56-57
PUBLIC keyword, 786
Publish view (NetObjects Fusion), 755-756
publishing
NetObjects Fusion, 755-756
Web Publishing wizard, 740-741
pull (client), 854
see also push
PULS (PowWow User Location Server), 673
pure HTML, 814, 821
advantages, 816
feedback, 816
platform independence, 816
support, 815
disadvantages, 816-817
availability of tags, 817
creativity loss, 817
extension workarounds, 822-823
frames, 822
multimedia files, 822-823
requirements for, 815
versus extensions, 820
push (server), 854
see also pull

Q

Q element, 83
QUANTITY, HTML 4.0 declaration, 54
QUERY STRING environment variable, 522, 540
question marks (?), 60
URL encoding, 520
QuickTime, 854
quotation marks (" ")
language-specific markup, 808-809
quotations, <BLOCKQUOTE> tag, 166-167

R

radio buttons, 282
Radio control, 603
RAM, 854
random numbers, generating, 559-560
random-access memory (RAM), 854
rangeFromElement( ) method, 420
raster graphics, 854
Ray Dream Designer, 727
system requirements, 727
Web site, 727
ray tracers, 727
POV-Ray, 727
system requirements, 727
Ray Dream Designer, 727
system requirements, 727
RCDATA content model, 59
reading machines, 6-7
RealAudio, 249, 661-662, 854
RealVideo, 656-657, 726
Web site, 726
reason fields (HTTP headers), 526-529
redirection response codes, 526
referencing
HTML controls, in frames, 604-605
objects
Document, 601
Elements, 603-604
Form, 601-603
Window, 601
styles, 946
Referer header, 96, 533
rel attribute
<A> tag, 227
<LINK> tag, 77, 145, 325
relative links, 224-225
<BASE> tag, 225-226
relative size (fonts), 949
size attribute, 175
relative URLs, 103
relative widths (tables), 456, 464
REMOTE ADDR environment variable, 540
REMOTE HOST environment variable, 540
REMOTE IDENT environment variable, 541
remote projects (Backstage Manager), 744

IPR2024-00143
Apple EX1011 Page 526

remote site administration (Backstage Manager), 743
REMOTE USER environment variable, 540
reports
Developer/2000 report writing module, 684
InContext WebAnalyzer, 702
Request for Comments (RFC), 854
request headers, 524-525, 846
determining, 524
environment variables based on, 538
AUTH TYPE, 540
CONTENT-LENGTH, 541
CONTENT-TYPE, 541
HTTP REQUEST METHOD, 538
PATH, 539
PATH INFO, 539
PATH TRANSLATED, 539
QUERY STRING, 540
REMOTE ADDR, 540
REMOTE HOST, 540
REMOTE IDENT, 541
REMOTE USER, 540
SCRIPT FILENAME, 539
SCRIPT NAME, 539
SERVER PORT, 538
SERVER PROTOCOL, 538
Full Method, 530
HTTP version, 532-534
method types, 530
Requested URL, 531
GET method, 530-531
proxy, 531
Method, 529-530
REQUEST METHOD environment variable, 521
requests (HTTP), 91-93
0.9, 93
1.0, 93
Authorization header field, 95
From header field, 95
GET method, 94
HEAD method, 94
header fields, 95
If-Modified-Since header field, 96
methods, 93
POST method, 94
Referer header field, 96
Request-Line, 93-94
Request-URI, 94-95
User-Agent header field, 96
#REQUIRED keyword, 64
reserved names, HTML 4.0 declaration, 53
reset buttons, 283
Reset control, 603
resize method, 354
resolutions, table measurements, 201
resources (on HTML), 960-966
browser technology sites, 962-964
Amaya, 963
Arena, 963
Browserwatch, 962
Cyberdog, 963
Galahad, 963
Internet Explorer, 963
Lynx, 963
NCSA Mosaic, 964
Netscape, 962
Opera, 964
SlipKnot, 964
editors/tools, 964-966
Backstage Internet Studio, 966
BBEdit, 964
HomeSite, 964
HotDog, 965
HoTMetaL Pro, 965
Microsoft FrontPad, 965
Microsoft FrontPage 97, 965
Netscape Composer, 965
PageMill, 966
Webber, 965
HTML Guru site, 962
HTML Writers Guild site, 962
Microsoft Site Builder Network site, 960
Netscape DevEdge OnLine site, 960
newsgroups, 966
W3C site, 960-961
Cougar, 961
CSS, 961
Dynamic HTML, 961
HTML 2 specification, 961
HTML 3.2 specification, 961
URL descriptions, 961
Web Design Group site, 962
Yahoo! site, 961
response headers, 525, 534-537
Content-Length, 536
Content-Type, 525, 536
Date, 535
enclosed URL, 536-537
Last Modified, 536
Server, 536
Set-Cookie, 536
status codes, 525-529
client error, 526
informational, 526
reason fields, 526-529
redirection, 526
server error, 526
success, 526
Status Response line, 534
responses (HTTP), 91, 96-99
Date header field, 98
Full-Response, 97
Location field, 99

Pragma header field, 98
Server field, 99
Status-Line, 97
Reason-Phrase, 97
Status-Code, 97
Retry-After header, 533
rev attribute
<A> tag, 227
<LINK> tag, 77, 145, 325
RFC (Request for Comments), 854
RGB color model, 447
defining color, 175
right value, clear attribute, 161
robots, 854
spiders, 856
root elements, XML, 787
Rosenbluth International Web site, see Travelution Web site
rows
aligning, 205
defining, 205
dynamically generating/sorting, 428-430
listing, 428-429
spanning (listing), 211
rows attribute, <FRAMESET> tag, 341
rowspan attribute, <TD> and <TH> tags, 209
rules, see lines

S

<SAMP> tag, 83, 169, 867
samplelayout.htm (listing), 627
Sams.net Web site, 706
San Diego Supercomputer Center Web site, 632
sans serif fonts, 371, 948
versus serif, 372
Sausage Software Web site, 698
Save Results Web bot, 738
scenes (VRML), 635
interactivity, 647-648
multiuser worlds, 648-649
scheme-specific-parts, 105-106
schemes, 106-108
file, 106
ftp, 106
gopher, 107
HTTP, 107
mailto, 107
news, 107
telnet, 108
see also protocols
SCRIPT FILENAME environment variable, 539
SCRIPT NAME environment variable, 539
Script Wizard, 610-613
<SCRIPT> tag, 79-80, 327, 553-555, 593-596, 871
attributes, 80
TYPE, 80
browser support, 80, 153
versus CGI, 80
disadvantages, 80
language attribute, 327
NOSCRIPT element, 80
src attribute, 327
type attribute, 327
W3C HTML 3.2 specification, 896
scripts, 326-328, 855
ActiveX
Script Wizard, 610-613
security, 614
attaching to HTML controls, 596-598
versus CGI, 80
editing, Backstage Designer, 749
executing
automatically, 594-595
on events, 595
subroutines, 595-596
Java, 579-582
JavaScript, 848
browser controls, 561
embedding, 555-556
form validation scripts, 557-559
hiding, 555
HTML markup, 553-555
layers, 358-361
random numbers scripts, 559-560
status bar messages, 560-561
SCRIPT element, 327
VBScript, 858
HTML markup, 593-598
scrollamount attribute, <MARQUEE> tag, 258
scrollbar controls, 607
scrolldelay attribute, <MARQUEE> tag, 258
scrolling, 855
scrolling attribute
<FRAME> tag, 342
<IFRAME> tag, 330
scrolling marquees, 257-258
aligning, 258
applets, 258
background color, 258
behaviors, 257
buffers, 258
delay (setting), 258
direction, 258
formatting text, 257
loops, 258
size, 258
search engines, 481-483
advantages, 481
customizing output, 482-483
keyword search box (adding), 482
parameters, 483
Web server support, 482
searching, 855
spiders, 856
Secured HyperText Transfer Protocol (SHTTP), 855

IPR2024-00143
Apple EX1011 Page 527

security
ActiveX, 614
encryption, 843
firewalls, 845
FrontPage, 733-734
administrator permissions, 734
author permissions, 734
end user permissions, 733
screening permissions, 734
Java, 570
byte code verification, 570
file access, 570
memory layout, 570
viruses, 859
Select control, 603
<SELECT> tag, 285-286, 875
W3C HTML 3.2 specification, 918-919
selection boxes (forms), 285-286
selectors, 946
classes, 946
contextual, 946
first-letter values, 946
first-line values, 946
separating, 946
semicolons (;), style sheets, 946
Sequential Query Language, see SQL
serif fonts, 371, 948
versus sans-serif, 372
SERVER ADMIN environment variable, 537
server error response codes, 526
Server header, 99, 534-536
SERVER NAME environment variable, 538
SERVER PORT environment variable, 538
SERVER PROTOCOL environment variable, 538
SERVER SOFTWARE environment variable, 538
server-side, 855
components, 606
image maps, 241, 264-268
including, 265
MAP files (creating), 266-268
script file path, 265
serverless video, 726
servers
administration, Backstage, 750
CERN, MAP files, 266-267
defined, 855
environment variables based on, 537
GATEWAY INTERFACE, 537
SERVER ADMIN, 537
SERVER NAME, 538
SERVER SOFTWARE, 538
glyph, 810
NCSA, MAP files, 267-268
Personal Web Server, 739
download site, 739
installing, 739
managing, 739
number of users, 739
proxy, 854
push, 854
sending data to
decoding data, 521-524
encoded data listing, 517
GET method, 516
name/value pairs, 518
path information, 518-519
registration form listing, 516-517
URL encoding, 519-521
Web
Apache, 772-773
Backstage Server add-on, 749-751
CERN httpd, 773
compatibility, 763-764
cost, 766
DBMS, 763
DNS, 763
FTP, 763
future needs, 765-766
Gopher, 763
HTTP, 763
IIS, 773-774
ISPs, 767-768
iWORLD Web site, 772
NCSA, 774
Netcraft Web site, 771
Netra j, 776
Netscape, 774-775
NNTP, 763
O'Reilly WebSite, 756
operating systems, 769-771
platform requirements, 764
platform/server combinations, 768
purposes, 761-762
resources, 768
scalability, 764
selecting, 761-762
SMTP, 763
software, 771-772
Telnet, 763
WebSite, 775
WebSite Pro, 775
WebSTAR, 776
SET command, 604
Set-Cookie header, 536
SGML (Standard Generalized Markup Language), 18, 856
applications
HTML as, 893-894
defining, 39-40
descriptive markup, 36-39
history, 39
1986 specification, 39
HTML 3.2 declaration, code listing, 927-929
HTML 4.0 declaration, 40-55
CAPACITY section, 51
character set declaration, 52

CHARSET section, 41-50
delimiters, 53
FEATURES section, 54-55
function characters, 52
naming rules, 53
quantity limits, 54
reserved names, 53
shunned characters, 51-52
SYNTAX section, 51-54
importance in HTML development, 66
limitations, 18
procedural markup, 36-39
syntax, 40
shape attribute, <AREA> tag, 269
Shape node, geometry field, 641
shapes attribute, <OBJECT> tag, 334
ShockWave, 259, 665-666, 722, 855
shopping carts, 855
SHTTP (Secured HyperText Transfer Protocol), 855
shunned characters
HTML 4.0 declaration, 51-52
Silicon Graphics
Cosmo Player plug-in, 667
Web site, 632
site maps, 478-481
color-coding, 481
Indian government example, 480
Lycos Web site, 479
maintenance, 481
text-only, 478
Travelution Web site, 494
Site view
(NetObjects Fusion), 752
sites (Web), see Web sites
size attribute
<AREA> tag, 269
<BASEFONT> tag, 84
<FONT> tag, 85, 173-175
text fields, 280
slashes (/)
trailing, 104
URL encoding, 520
XML, 785
SlipKnot Web site, 964
small-caps fonts, 948
<SMALL> tag, 84, 176, 868
Smartdubbing software, 668
SMTP Web servers, 763
sockets, 855
software
interface elements, 443
translation software, 671
Web servers, 771-772
see also plug-ins
Sony Web site, 132
sound plug-ins, 659-663
source editors, 692
sources, <CITE> tag, 167
space (page layout), 293
spacer images (layout), 460-461
spaces, trailing (ordered lists), 189
spacing
letters, 951
line, <BR> tag, 160
table cells, 203
words, 951
<SPAN> tag, 82, 367, 403
CSS, 326
listing, 403
spanning, 209
columns (listing), 209-210
rows (listing), 211
special characters (list), 882-889
SPG Web site, 717
SphereSensor node, 637
spiders, 856
spinbutton control, 607
sprites, 718
SQL (Sequential Query Language), 855
SQL Server, 684
publishing technologies, 684
Web Assistant, 684
Web site, 684
src attribute
<FRAME> tag, 342
<IFRAME> tag, 329
<IMG> tag, 235
<SCRIPT> tag, 327
Standard Generalized Markup Language, see SGML
standards (HTML), versus extensions, 112-113
horizontal rules listing, 112
standby attribute, <OBJECT> tag, 334
start attribute, <OL> tag, 184-186
start method, 576
starting Java applets, 576
stateless protocols, 91
status bars (messages), 560-561
Stenbrink, Susan (Travelution Web site), 487
stop method, 576
stopping Java applets, 576
streaming, 856
video, 653, 656, 726
RealVideo, 726
VivoActive, 726
strike-through text, 171-172
<STRIKE> tag, 84, 171-172, 868
<STRONG> tag, 83, 164-165, 867
structural elements
<BASE> tag, 143-144
HREF attribute, 143
<BODY> tag, 148-152
BACKGROUND attribute, 149
<!DOCTYPE> tag, 139
<HEAD> tag, 140-141
listing, 141
<HTML> tag, 139-140
advantages, 140
listing, 140
<ISINDEX> tag, 144-145
prompt attribute, 144

IPR2024-00143
Apple EX1011 Page 528

<LINK> tag, 145-146
href attribute, 145
name attribute, 145
rel attribute, 145
rev attribute, 145
title attribute, 145
<META> tag, 146-148
hppt-equiv attribute, 146
name attribute, 146-147
<SCRIPT> tag, 152-153
browser support, 153
<STYLE> tag, 148
<TITLE> tag, 141-143
listing, 142
structure (documents), 36, 856
style attribute, 326
<BUTTON> tag, 332
<FIELDSET> tag, 333
JavaScript style sheets, 403
<LABEL> tag, 331
style sheets, 386
attributes
background, 951
background-attachment, 950
background-color, 950
background-image, 950
background-position, 950-951
background-repeat, 950
borders, 954-955
display, 956
font, 949
font-family, 948
font-size, 949
font-style, 948
font-variant, 948
font-weight, 948-949
letter-spacing, 951
line-height, 953
list-style, 957
list-style position, 957
list-style-image, 957
list-style-type, 956
margins, 953
padding, 953-954
text boxes, 955
text-align, 952
text-decoration, 951
text-indent, 952
text-transform, 952
vertical-align, 952
white-space, 956
word-spacing, 951
backgrounds, 379-380
color, 379
graphics, 379
splashes, 379
cascading, 323-326, 841
DIV element, 326
implementation, 324
LINK element, 325
SPAN element, 326
style control, 364-366
STYLE element, 325-326
W3C recommendation, 961
combining, 370
cascading order, 370
embedded, 366-368, 947
format, 947
syntax, 366
text style attributes, 374-376
examples of
business page, 386-387
employee newsletter, 388-390
Kid's News, 395-397
online book cover, 397-398
party invitation, 394-395
price list, 390-393
grouping, 381
assigning classes, 381-382
attributes, 381
tags, 381
HTML level 3, 26
inline, 366-368, 947
JavaScript, 400-401, 848
backgroundImage property, 409
block-level element properties, 407-408
browser support, 401
class attribute, 403
classes, 401
color property, 409
display property, 409
font properties, 404-405
ids attribute, 403
<LINK> tag, 402
listStyleType property, 409
precedence order, 410
<SPAN> tag, 403
style attribute, 403
<STYLE> tag, 402
text properties, 405-406
whiteSpace property, 409
layout style attributes, 376-378
margin-left, 376
margin-right, 376
margin-top, 376
text-align, 376
text-indent, 376
linked, 366, 369, 390, 947
format, 947
syntax, 366
proposals, 27
STYLE element, 81
browser support, 81
syntax, 946-947
classes, 946
commas (,), 946
curly brackets ({}), 946
selectors, 946
semicolons (;), 946
text style attributes, 371-376
font size, 372
font-family, 371
font-style, 373
font-weight, 373
line-height, 374
text-decoration, 373
TITLE element, 81
Style view (NetObjects Fusion), 754

<STYLE> tag, 81, 148, 402, 419, 947
attributes
media, 326
title, 326
type, 81, 326
browser support, 81
CSS, 325-326
listing, 402
W3C HTML 3.2 specification, 896
styles
attributes, 947
background, 951
background-attachment, 950
background-color, 950
background-image, 950
background-position, 950-951
background-repeat, 950
borders, 954-955
display, 956
font, 949
font-family, 948
font-size, 949
font-style, 948
font-variant, 948
font-weight, 948-949
letter-spacing, 951
line-height, 953
list-style, 957
list-style position, 957
list-style-image, 957
list-style-type, 956
margins, 953
padding, 953-954
text boxes, 955
text-align, 952
text-decoration, 951
text-indent, 952
text-transform, 952
vertical-align, 952
white-space, 956
word-spacing, 951
borders, 955
logical, 163-164
<ADDRESS> tag, 165
<BLOCKQUOTE> tag, 166-167
<CITE> tag, 167
<CODE> tag, 167-168
<DFN> tag, 167
<EM> tag, 164-165
<KBD> tag, 169
review, 170
<SAMP> tag, 169
<STRONG> tag, 164-165
logical list, 194
physical, 170
<B> tag, 171
<FONT> tag, 173-176
<I> tag, 171
<STRIKE> tag, 171-172
<SUB> tag, 172-173
<SUP> tag, 172-173
<TT> tag, 172
<U> tag, 171
word processing, 163
referencing, 946
text, 157
<SUB> tag, 84, 172-173, 868
submit buttons, 282-283
Submit control, 603
subscript text, 172-173
success response codes, 526
Sun Microsystems Web site, 776
<SUP> tag, 84, 172-173, 868
superscript text, 172-173
surfing, 856
Surround Video, 666
Sybase, 683-684
NetImpact Dynamo, 683
gateways supported, 683
HTML authoring tools, 683
Web site, 683
Web site, 683
web.sql, 683
SYNTAX section (HTML 4.0 declaration), 51-54
character set declaration, 52
delimiters, 53
function characters, 52
naming rules, 53
quantity limits, 54
reserved names, 53
SCOPE declaration, 51
shunned characters, 51-52
SYSTEM keyword, 786

T

tabindex attribute, <BUTTON> tag, 332
<TABLE> tag, 82, 876-877
attributes
align, 199-202
border, 203-204
cellpadding, 203-204
cellspacing, 203-204
width, 199-202
constraints, 198-199
W3C HTML 3.2 specification, 909-913
tables, 856
aligning, 199-202
appearances, 203-204
attribute precedence, 213-214
browser incompatibility, preformatted text, 162
building, 3×3 table listing, 209
<CAPTION> tag, 204
cells, 841
CGI table example listing, 215-216
column and headline example listing, 217-218
constraints, 198-199
HTML 3.2 W3C specification, 909-913
HTML 4, 335, 456-457

IPR2024-00143
Apple EX1011 Page 529

HTML table model Web site, 943
measuring systems, choosing, 201
page layout, 456-457, 461-465
absolute widths, 456
CyberMad Web site, 504-505
guidelines, 462-463
nested tables, 464-465
newspaper columns, 463-464
relative widths, 456, 464
Travelution Web site, 492
see also cells; rows
tabstrip control, 607
tags
<!-- -->, 40, 862
<A>, 223, 865-866
href attribute, 223-224
internal anchors, 229-232
mailto: protocol, 226-227
name attribute, 228
target attribute, 226, 343-344
title attribute, 227-228
W3C HTML 3.2 specification, 920
<ADDRESS>, 165, 869
W3C HTML 3.2 specification, 901
<APPLET>, 248, 578-579, 872
syntax, 253-254
W3C HTML 3.2 specification, 923-925
<AREA>, 269, 873
coords attribute, 269-270
shape attribute, 269
size attribute, 269
<B>, 171, 867
<BASE>, 143-144, 863
HREF attribute, 143
relative links, 225-226
W3C HTML 3.2 specification, 897
<BASEFONT>, 84, 869
size attribute, 175
W3C HTML 3.2 specification, 926
<BGSOUND>, 257, 871
<BIG>, 176, 868
<BLINK>, 869
block formatting elements, 82
<BLOCKQUOTE>, 166-167, 869
W3C HTML 3.2 specification, 908
<BODY>, 148-152, 862
BACKGROUND attribute, 149
W3C HTML 3.2 specification, 899-900

, 159-161, 868
W3C HTML 3.2 specification, 926
<BUTTON>, 332
class attribute, 332
dir attribute, 332
disabled attribute, 332
id attribute, 332
lang attribute, 332
style attribute, 332
tabindex attribute, 332
title attribute, 332
<CAPTION>, 204, 877
W3C HTML 3.2 specification, 911
<CENTER>, 869
W3C HTML 3.2 specification, 907
<CITE>, 167, 867
<CODE>, 167-168, 867
<COMMENT>, 862
defined, 23, 856
<DD>, 192, 867
<DFN>, 167, 867
<DIR>, 194, 866
W3C HTML 3.2 specification, 906
<DIV>, 163, 367, 865
CSS, 326
W3C HTML 3.2 specification, 907
<DL>, 192-194, 867
W3C HTML 3.2 specification, 905
<!DOCTYPE>, 139
<DT>, 192, 867
versus elements, 73
<EM>, 164-165, 867
<EMBED>, 660, 872-873
plug-ins, 260
emphasis, 164
<FIELDSET>, 333
class attribute, 333
dir attribute, 333
id attribute, 333
lang attribute, 333
style attribute, 333
title attribute, 333
<FIG>, 270
<FONT>, 173, 869
color attribute, 175
size attribute, 173-175
W3C HTML 3.2 specification, 925
<FORM>, 276-277, 874
accept-charset attribute, 803-804
ACTION field, 518, 521
CGI programs, 514-515
enctype attribute, 277
GET method, 276
method attribute, 276
POST method, 276-277
W3C HTML 3.2 specification, 908
<FRAME>, 341-342, 881-882
frameborder attribute, 342
listing, 341
marginheight attribute, 342
marginwidth attribute, 342

name attribute, 342
noresize attribute, 342
scrolling attribute, 342
src attribute, 342
<FRAMESET>, 340-341, 881
border attribute, 340
cols attribute, 340
frameborder attribute, 340
rows attribute, 341
grouping, style sheets, 381
<H1> - <H6>, 156-157, 865
<HEAD>, 76, 140-141, 862
listing, 141
W3C HTML 3.2 specification, 895
<HR>, 176, 868
W3C HTML 3.2 specification, 908-909
<HTML>, 139-140, 862
advantages, 140
listing, 140
<I>, 171, 867
<IFRAME>, 329-330
align attribute, 330
frameborder attribute, 329
height attribute, 330
listing, 330
marginheight attribute, 329
marginwidth attribute, 329
name attribute, 329
scrolling attribute, 330
src attribute, 329
width attribute, 330
<ILAYER>, 349
above attribute, 350
background attribute, 350
below attribute, 350
bgcolor attribute, 350
clip attribute, 350
left attribute, 351
name attribute, 351
top attribute, 351
visibility attribute, 351
width attribute, 351
zindex attribute, 351
<IMG>, 234-235, 847, 869-871
align attribute, 235-238
alt attribute, 235
border attribute, 240
dynsrc attribute, 256-257
height attribute, 239
hspace attribute, 240
ismap attribute, 241
src attribute, 235
usemap attribute, 242, 269
vspace attribute, 240
W3C HTML 3.2 specification, 920-923
width attribute, 239
<INPUT>, 278-283, 596-598, 874-875
CHECKBOX input type, 281
HIDDEN input type, 281
IMAGE input type, 283
PASSWORD input type, 280
RADIO input type, 282
RESET input type, 283
SUBMIT input type, 282-283
TEXT input type, 279-280
W3C HTML 3.2 specification, 915-918
<ISINDEX>, 144-145, 863
prompt attribute, 144
W3C HTML 3.2 specification, 896
<KBD>, 169, 867
<LABEL>, 331-332
accesskey attribute, 332
class attribute, 331
dir attribute, 332
disabled attribute, 332
for attribute, 331
id attribute, 331
lang attribute, 332
onBlur attribute, 332
onClick attribute, 332
onFocus attribute, 332
style attribute, 331
title attribute, 332
<LAYER>, 349, 416
above attribute, 350
background attribute, 350
below attribute, 350
bgcolor attribute, 350
clip attribute, 350
left attribute, 351
methods, 353-354
name attribute, 351
top attribute, 351
visibility attribute, 351
width attribute, 351
zindex attribute, 351
<LI>, 181-182, 866-867
<LINK>, 77, 145-146, 402, 947, 863-864
CSS, 325
href attribute, 145, 325
listing, 402
media attribute, 325
name attribute, 145
rel attribute, 145, 325
rev attribute, 145, 325
title attribute, 145, 325
type attribute, 325
W3C HTML 3.2 specification, 898-899
<LISTING>, 163
logical, 164
<MAP>, 269-271, 873
W3C HTML 3.2 specification, 926-927
<MARQUEE>, 257-258
align attribute, 258
behavior attribute, 257
bgcolor attribute, 258
direction attribute, 258
height attribute, 258
hspace attribute, 258
loop attribute, 258

IPR2024-00143
Apple EX1011 Page 530

scrollamount attribute, 258
scrolldelay attribute, 258
vspace attribute, 258
width attribute, 258
<MENU>, 194, 866
W3C HTML 3.2 specification, 906
<META>, 78, 146-148, 864
hppt-equiv attribute, 146
name attribute, 146-147
W3C HTML 3.2 specification, 897
<NEXTID>, 864
<NOBR>, 868
<NOEMBED>, 873
<NOFRAME>, 882
<OBJECT>, 333-334, 871
ActiveX controls, 249, 261
align attribute, 334
alt attribute, 334
applets, 248
border attribute, 334
CLASSID attribute, 248
classid attribute, 333
codebase attribute, 333
CODETYPE attribute, 248
codetype attribute, 334
data attribute, 333
declare attribute, 333
height attribute, 334
hspace attribute, 334
id attribute, 333
level 4 expansion, 247
name attribute, 334
plug-ins, 248
shapes attribute, 334
standby attribute, 334
title attribute, 334
type attribute, 333
usemap attribute, 334
vspace attribute, 334
width attribute, 334
<OL>, 866
start attribute, 184-186
type attribute, 187-189
W3C HTML 3.2 specification, 904-905
<OPTION>, 875
<P>, 158-159, 865
W3C HTML 3.2 specification, 903
<PARAM>, 334-335, 579, 872
name attribute, 335
value attribute, 335
valuetype attribute, 335
<PLAINTEXT>, 163
<PRE>, 161-162, 868
W3C HTML 3.2 specification, 906-907
<SAMP>, 169, 867
<SCRIPT>, 79, 152-153, 327, 553-555, 593-596, 871
browser support, 153
language attribute, 327
src attribute, 327
type attribute, 327
W3C HTML 3.2 specification, 896
<SELECT>, 285-286, 875
W3C HTML 3.2 specification, 918-919
<SMALL>, 176, 868
<SPAN>, 367, 403
CSS, 326
listing, 403
<STRIKE>, 171-172, 868
<STRONG>, 164-165, 867
<STYLE>, 148, 402, 419, 947
CSS, 325-326
listing, 402
media attribute, 326
title attribute, 326
type attribute, 326
W3C HTML 3.2 specification, 896
<SUB>, 172-173, 868
<SUP>, 172-173, 868
<TABLE>, 198-199, 876-877
align attribute, 199-202
border attribute, 203-204
cellpadding attribute, 203-204
cellspacing attribute, 203-204
width attribute, 199-202
W3C HTML 3.2 specification, 909-913
<TD>, 208, 879-880
align attribute, 212
colspan attribute, 209
height attribute, 209
nowrap attribute, 212
placement conventions, 208
rowspan attribute, 209
valign attribute, 212
W3C HTML 3.2 specification, 911
width attribute, 208
<TEXTAREA>, 284, 875
W3C HTML 3.2 specification, 919
<TH>, 207, 878-879
align attribute, 212
colspan attribute, 209
height attribute, 209
nowrap attribute, 212
rowspan attribute, 209
valign attribute, 212
W3C HTML 3.2 specification, 911
width attribute, 208
<TITLE>, 81, 141-143, 865
listing, 142
W3C HTML 3.2 specification, 895-896
<TR>, 205, 878
align attribute, 205
valign attribute, 205
W3C HTML 3.2 specification, 911

<TT>, 172, 867
<U>, 171
<UL>, 189-190, 866
bullets, 190-192
W3C HTML 3.2 specification, 903-904
<VAR>, 168, 867
<WBR>, 868
XML, 785
<XMP>, 163,
see also elements
target attribute, <A> tag, 226, 343-344
TCP (Transport Control Protocol), 857
TCP/IP (Transfer Control Protocol/Internet Protocol), 88, 90-91, 857
history, 89-90
ftp, 89
telnet, 89
TCP, 91
demultiplexing, 91
<TD> tag, 208, 879-880
attributes
align, 212
colspan, 209
height, 209
nowrap, 212
rowspan, 209
valign, 212
width, 208
placement conventions, 208
W3C HTML 3.2 specification, 911
teleconferencing plug-ins, 672-674
teletype text, see monospaced text
Telnet, 89, 856
URLs, 108
Web servers, 763
templates, 301-308, 856
banners, 302-303
business page, 386-387
children's page, 395-397
content area, 303
creating pages from, 306-308
adding lists, 311-312
home pages, 308-311
listing, 306-307
employee newsletter, 388-390
Explorer, 733
Customer Support Web, 733
Empty Web, 733
Normal Web, 733
Personal Web, 733
Project Web, 733
FrontPage Editor, 736
online book cover, 397-398
party invitation, 394-395
price list, 390-393
structure listing, 301
template.htm file listing, 305
terminals, 857
terms, defining, 167
alignment, 376
baseline, 952
bottom, 952
cell content, 212
middle, 952
percentage, 952
sub, 952
super, 952
text-bottom, 952
text-top, 952
top, 952
attributes
letter-spacing, 951
line-height, 953
text-align, 952
text-decoration, 951
text-indent, 952
text-transform, 952
vertical-align, 952
word-spacing, 951
bidirectional, 807
<BIG> tag, 176
borders, attributes, 954-955
capitalization, 952
color
color attribute (<FONT> tag), 175
text attribute, 150-151
colors, changing, 176
dynamic, 417-420
code listing, 418
enabling, 420
em spaces, 951
formatting, 158
Backstage Designer, 746-747
*
 tag, 159-161*
by usage, 163-164
<DIV> tag), 163
headings, 156-157
marquees, 257
ordered lists, 183
<P> tag, 158-159
preformatted, 161-163
indenting, 376, 952
JavaScript style sheets, 405-406
lineHeight property, 405
textAlign property, 406
textDecoration property, 405
textIndent property, 406
textTransform property, 406
verticalAlign property, 405-406
justification, 952
language-specific markup, 807-809
alignment, 809
cursive joining behavior, 808
hyphenation, 809
quotation marks, 808-809
writing direction, 807-808
layout, 453-454
leading, 849
logical styles, 163
address info, 165
emphasis, 164-165

IPR2024-00143
Apple EX1011 Page 531

program code, 167-169
quotations, 166-167
review, 170
term definition, 167
margins, attributes, 953
markup, 850
logical, 849
padding, attributes, 953-954
physical styles, 170
bold, 171
fonts, 173-176
italic, 171
monospaced, 172
strike-through, 171-172
subscripts, 172-173
superscripts, 172-173
underline, 171
preformatted, HTML 3.2 W3C specification, 906-907
<SMALL> tag, 176
sources, 167
style attributes, 371-376
embedded style sheets, 374-376
font size, 372
font-family, 371
font-style, 373
font-weight, 373
line-height, 374
text-decoration, 373
styles, 157
table alignment
see also fonts
text areas (forms), 284
text attribute, <BODY> tag, 150-151
text boxes, 953
attributes
border, 954-955
clear, 955
float, 955
height, 955
margin, 953
padding, 953-954
width, 955
Text control, 603
text elements, 857
text fields, 279-280
maxlength attribute, 280
size attribute, 280
value attribute, 280
text-align attribute, 376, 952
text-decoration attribute, 373, 951
values, 951
text-indent attribute, 376, 952
text-level elements, W3C HTML 3.2 specification, 913
text-transform attribute, 952
textAlign property
JavaScript style sheets, 406
Label control, 609
<TEXTAREA> tag, 284, 875
W3C HTML 3.2 specification, 919
textbox control, 607
textDecoration property, JavaScript style sheets, 405
textIndent property, JavaScript style sheets, 406
textTransform property, JavaScript style sheets, 406
textures (VRML), 637, 641
<TH> tag, 207, 878-879
attributes
align, 212
colspan, 209
height, 209
nowrap, 212
rowspan, 209
valign, 212
width, 208
W3C HTML 3.2 specification, 911
themes (Web sites), 439
audience, 440-441
content, 441
look and feel of site, 441-442
site goals, 439-440
Novartis site, 440
thin clients, 851, 857
third-party ActiveX controls, 607
Ticker applet, 586-587
Timer node, 637
title attribute
<A> tag, 227-228
<BUTTON> tag, 332
<FIELDSET> tag, 333
<LABEL> tag, 332
<LINK> tag, 145, 325
<OBJECT> tag, 334
<STYLE> tag, 326
Title header, 534
<TITLE> tag, 81, 141-143, 865
listing, 142
W3C HTML 3.2 specification, 895-896
titles, 141-143
multiple, 143
size, 142
To Do list (Front Page), 738
toggle button control, 607
toolbars, 857
tools (Web authoring)
Backstage, 742-751
FrontPage, 730-741
NetObjects Fusion, 751-756
top attribute, <LAYER> and <ILAYER> tags, 351
TouchSensor node, 637, 643
<TR> tag, 205, 878
attributes
align, 205
valign, 205
W3C HTML 3.2 specification, 911
trailing slashes, 104
trailing spaces (ordered lists), 189
transactions, HTTP, 91
Transfer Control Protocol/ Internet Protocol, see TCP/IP
Transform node, 638
rotation field, 642
scale field, 642
translation field, 642

transformations (VRML), 637
translation software, 671
Transport Control Protocol, see TCP
Transport Control Protocol/ Internet Protocol, see TCP/IP
Travelution Web site, 486-498
interface design, 488-492
consistency, 489
hyperlink code, 492
location hints, 489
spawning code, 492
navigation, 494-496
image map, 494
site map, 494
page layout, 492-494
animated GIFs, 493
flaw, 493
fonts, 492
graphics, 492
tables, 492
Planetcam, 497
Tribal Voice's PowWow plug-in, 672-673
<TT> tag, 84, 172, 867
Tucows service Web site, 706
type attributes
input fields, 915-917
<LINK> tag, 325
<OBJECT> tag, 333
<OL> tag, 187-189
<SCRIPT> tag, 80, 327
with start attribute, 189
<STYLE> tag, 326
<UL> tag, 190-192
type sizes, 174
type style (physical styles), 170
typefaces
headings, 157
subscripts, 172
superscripts, 172
typesetting, underlining, 171

U

<U> tag, 84, 171
UCS (Universal Multiple-Octet Coded Character Set), 798-799, 858
BMP (Basic Multilingual Plane), 799
Transformation Format, 857
<UL> tag, 189-190, 866
bullet type (type attribute), 190-192
W3C HTML 3.2 specification, 903-904
Ultimate Compendium of Photoshop Sites Web site, 714
underlining, 171
Unicode, 43, 857
Unicode Consortium Web site, 797
Uniform Resource Characteristic (URC), 108
format, 108
Uniform Resource Citation (URC), 857
Uniform Resource Identifiers, see URIs
Universal Multiple-Octet Coded Character Set, see UCS
UNIX, 858
animated GIF files (creating), 256
Web servers, 769-770
advantages, 769
disadvantages, 770
unordered lists
HTML 3.2 W3C specification, 903-904
type attribute, 192
<UL> tag, 189-190
bullet type (type attribute), 190-192
URC (Uniform Resource Citation), 857
URC (Uniform Resource Characteristic), 108
format, 108
URIs (Uniform Resource Identifiers), 14, 88, 102-104, 857
extensibility, 109
file paths, 103
Internet nodes, 103
optional arguments, 104
protocols, 103
relative URLs, 103
trailing slashes, 104
versus URLs, 88
URL encoding, 105, 519-521
decoding data, 521-524
CONTENT LENGTH environment variable, 522
converting special characters, 522
listing, 523-524
QUERY STRING environment variable, 522
REQUEST METHOD environment variable, 521
splitting name/value pairs, 522
percent signs (%), 519
reserved characters, 519-521
ampersands (&), 520
equal signs (=), 520
number signs (#), 521
percent signs (%), 521
question marks (?), 520
slashes (/), 520
resources, 521
steps, 521
URL-header header, 534
URLs, 104-108, 857
descriptions, W3C Web site, 961
format example, 104
loading to frames, 226
problems, 108

IPR2024-00143
Apple EX1011 Page 532

relative, 103
scheme-specific-parts, 105-106
schemes, 106-108
file, 106
ftp, 106
gopher, 107
HTTP, 107
mailto, 107
news, 107
telnet, 108
unsafe characters, 105
versus URIs, 88
URNs, 857
US ASCII, see 7-bit ASCII
usemap attribute
<IMG> tag, 242, 269
<OBJECT> tag, 334
Usenet newsgroups, 858
alt.lang.java, 589
HTML newsgroups, 966
user profiles, Backstage Manager, 743
User-Agent header, 96, 534
user-interface controls, 606
usernames, 858
utilities
Bandwidth Buster, 704-705
compression, 704
downloading, 704
interface, 705
LOW SRC images, 704
text-only sites, 705
Crunch!, 705-706
HTML Validator, 699-701
capabilities, 700
cost, 701
syntax error list, 700
tags required, 701
InContext WebAnalyzer, 701-702
display, 702
reports, 702
setting options, 702
Xpire Plus, 703-704
disadvantages, 704
New! icons, 704

V

valid documents, 858
XML, 785-786
validation, 858
Backstage Designer, 749
forms, 557-559
HTML Validator, 699-701
capabilities, 700
cost, 701
syntax error list, 700
tags required, 701
valign attribute
<TD> and <TH> tags, 212
<TR> tag, 205
value attribute
<LI> tag, 192
<PARAM> tag, 335
text fields, 280
values
parameters, 858
type attribute, 187
valuetype attribute, <PARAM> tag, 335
<VAR> tag, 83, 867
<CODE> tag, 168
variables, CGI environment, 537-541
AUTH TYPE, 540
CONTENT LENGTH, 522
CONTENT-LENGTH, 541
CONTENT-TYPE, 541
GATEWAY INTERFACE, 537
HTTP REQUEST METHOD, 538
PATH, 539
PATH INFO, 539
PATH TRANSLATED, 539
QUERY STRING, 522, 540
REMOTE ADDR, 540
REMOTE HOST, 540
REMOTE IDENT, 541
REMOTE USER, 540
request header-based, 538
REQUEST METHOD, 521
SCRIPT FILENAME, 539
SCRIPT NAME, 539
SERVER ADMIN, 537
SERVER NAME, 538
SERVER PORT, 538
SERVER PROTOCOL, 538
SERVER SOFTWARE, 538
server-based, 537
VBScript, 858
HTML markup, 593
<INPUT> tag, 596-598
<SCRIPT> tag, 593-596
HTML object model, 598
Document object, 601
Elements object, 603-604
Form object, 601-603
hierarchy, 599-600
variables, 604
Window object, 601
importance of, 592
vector graphics, 858
vector programs, 711-712, 719-722
Adobe Illustrator, 721
features, 721
system requirements, 721
advantages, 711, 719
common features, 720
CorelDRAW!, 720-721
features, 720
system requirements, 721
Corel Xara, 722
disadvantages, 712
file sizes, 711
Fractal Expression, 720
Macromedia FreeHand, 721-722
features, 722
Shockwave plug-in, 722
system requirements, 722
resolution loss, 711
vertical bars (|), 60

vertical-align attribute, 952
baseline value, 952
bottom value, 952
capitalize value, 952
lowercase value, 952
middle value, 952
percentage value, 952
sub value, 952
super value, 952
text-bottom value, 952
text-top value, 952
top value, 952
uppercase value, 952
verticalAlign property, JavaScript style sheets, 405-406
video, 723
Internet Explorer, 256-257
plug-ins, 654-659
Surround Video, 666
QuickTime, 854
serverless, 726
streaming, 653, 656, 726
RealVideo, 726
VivoActive, 726
technologies available, 723
video conferencing plug-ins, 672-674
VideoPhone 2.0, 674
viewers, 858
Java Applet Viewer, 572-573
Viewpoint node, 643
viewpoints, VRML, 637
Viola, 28, 858
Virtual Reality Modeling Language, see VRML
viruses, 859
visibility attribute, <LAYER> and <ILAYER> tags, 351
visited links, color, 152
VivoActive, 726
compression rates, 726
Web site, 726
VLINK attribute, <BODY> tag, 152
Voodoo JavaScript Tutorial Web site, 564
VREAM WIRL plug-in, 667
VRML (Virtual Reality Modeling Language), 631-632, 653, 858
adding to Web pages, 644
billiard table listing, 638-639
DEF keyword, 640
embedding in HTML, 644-645
billiard table listing, 644-645
events, 637
frames, 645-646
listing, 645
future developments, 647-649
interactivity, 647-648
multiuser worlds, 648-649
header line syntax, 640
hierarchy, 641
history, 633-634
Pesce, Mark, 633
VAG, 633
HTML and
communication, 646-647
differences, 636-637
similarities, 635-636
lighting, 637
nodes, 635
Appearance, 641
fields, 635
geometry, 636
Group, 638
Inline, 638
PositionInterpolator, 637
reusing, 643
SphereSensor, 637
Timer, 637
TouchSensor, 637
Transform, 638
plug-ins
Cosmo Player, 667
WIRL, 667
reference documents, 637
scenes, 635
textures, 637, 641
TouchSensor, 643
transformations, 637
version 1, 632
version 2, 632
viewers (CosmoPlayer), 632
viewpoints, 643, 637
VRML Consortium, 634
VRML Repository page, 632
Web site appearance, 632-633
viewers, 632
VRML Architecture Group (VAG) Web site, 633
VRML cafe Web site, 632
vspace attribute
<IMG> tag, 240
<MARQUEE> tag, 258
<OBJECT> tag, 334

W

W3C (World Wide Web Consortium), 25, 32-33, 859
charter, 32
HTML 3.2 specification, 892-942
A element, 920
ADDRESS element, 901
APPLET element, 923-925
BASE element, 897
BASEFONT element, 926
block elements, 902
BLOCKQUOTE element, 908
BODY element, 899-900
BR element, 926
CAPTION element, 911
CENTER element, 907
DIR element, 906
DIV element, 907
DL element, 905

IPR2024-00143
Apple EX1011 Page 533

document structure, 894
DTD, 929-940
FONT element, 925
font style elements, 913-914
FORM element, 908
form fields, 914
HEAD element, 895
headings, 901
HR element, 908-909
IMG element, 920-923
INPUT element, 915-918
ISINDEX element, 896
ISO Latin-1 character entities, 940-942
LINK element, 898-899
MAP element, 926-927
MENU element, 906
META element, 897
OL element, 904-905
P element, 903
phrase elements, 914
PRE element, 906-907
sample SGML open catalog, 927
SCRIPT element, 896
SELECT element, 918-919
SGML declaration, 927-929
STYLE element, 896
TABLE element, 909-913
TD element, 911
text-level elements, 913
TEXTAREA element, 919
TH element, 911
TITLE element, 895-896
TR element, 911
UL element, 903-904
HTML 4 enhancements, 321
Web site, 65, 320, 943, 946, 960-961
Cougar, 961
CSS, 961
Dynamic HTML, 961
HTML 2 specification, 961
HTML 3.2 specification, 961
URL descriptions, 961
W3C httpd Web server, see CERN httpd Web server
WAIS (wide area information service), 859
watermarks, 859
Wave, 131
WBL (Web Baseball League) example, 417
data source, wblteam.txt file listing, 430
data-aware pages, 423-430
binding to records, 424-428
data source control, 424
dynamic presentation listing, 424-426
features, 424
generating/sorting table rows, 428-430
dynamic graphics, 420-423
drag-and-drop listing, 421-422
ZINDEX values, 420
dynamic text, 417-420
enabling, 420
team page listing, 418
features, 417
<WBR> tag, 868
Web Baseball League example, see WBL example
Web bots, FrontPage, 738
capabilities, 738
inserting, 738
Save Results, 738
Web Design Group Web site, 962
Web Developer
requirements, 686
security, 686
Web pages
content, 842
data input
decoding data, 521-524
encoded data listing, 517
GET method, 516
name/value pairs, 518
path information, 518-519
registration form listing, 516-517
URL encoding, 519-521
data-aware, 423-430
binding to records, 424-428
data source control, 424
dynamic presentation listing, 424-426
features, 424
generating/sorting table rows, 428, 428-430
defined, 852
design
audience, 440-441
content, 441
interface, 436-439
look and feel of site, 441-442
site goals, 439-440
themes, 439
WebTV, 132
dynamic, 843
editing
NetObjects Fusion, 753
PageDraw, 756
exporting from desktop publishers, 466
hits, 846
layout, 292-301, 452
animated graphics, 455
backgrounds, 456
classic layout example (columns), 296-297
conventional layout example, 293-295
CSS, 454-455
CyberMad Web site, 503-507
elements, 453
fonts, 454
formal page example, 299-301
frames, 344-347

graphics, 455, 465-466
links, 457
modern layout example, 295
page dimensions, 457-458
plug-ins, 458
programmed controls, 457
space, 293
spacer images, 460-461
style attributes, 376-378
tables, 456-457, 461-465
technical information page example, 297-298
templates, 301-308
text, 453-454
Travelution Web site, 492-494
without tables, 458-461
loading audio, 457
multiple, 820
navigation
CyberMad Web site, 507-508
Travelution Web site, 494-496
nodes, 851
templates
Explorer, 733
FrontPage Editor, 736
updating, Xpire Plus utility, 703-704
VRML, 644
wizards
Explorer, 733
FrontPage Editor, 736-737
Web Publishing Wizard, 740-741
connection information, 741
providers supported, 740
Web, see WWW
Web servers
Apache, 772-773
Apache Development Group, 772
Cygnus Web site, 773
mailing list, 773
technical support, 773
Backstage Server add-on, 749-751
requirements, 750
CERN httpd, 773
features, 773
compatibility, 763-764
cost, 766
personnel, 766
DBMS, 763
DNS, 763
FTP, 763
future needs, 765-766
Gopher, 763
HTTP, 763
IIS, 773-774
disadvantages, 774
technical support, 774
ISPs, 767-768
iWORLD Web site, 772
NCSA, 774
bugs, 774
downloading, 774
Netcraft Web site, 771
Netra j, 776
cost, 776
Netscape, 774-775
cost, 774
evaluation copies, 774
technical support, 775
NNTP, 763
O'Reilly WebSite, 756
operating systems, 769
MacOS, 770
OS/2, 771
UNIX, 769-770
Windows, 770
platform requirements, 764
platform/server combinations, 768
purposes, 761-762
resources, 768
scalability, 764
selecting, 761-762
SMTP, 763
software, 771-772
Telnet, 763
WebSite, 775
cost, 775
O'Reilly and Associates Web site, 775
technical support, 775
WebSite Pro, 775
WebSTAR, 776
cost, 776
trial copies, 776
Web sites
Accent Software, 670
Access 97, 682
Adobe, 693
Adobe Illustrator, 721
Adobe Photoshop, 713
Alchemy Mindworks GIF Construction Set, 256
Allaire Corp, 694
Alta Vista, 481
Amaya, 963
Andy's Photoshop tips, 715
Animator applet, 582
Apache Development Group, 772
Arena, 963
audience, 440-441
Backstage Internet Studio, 966
BBEdit, 695, 964
Beginner's Guide to JavaScript, 563
Bitstream, 810
Black Diamond, 666
Browserwatch, 962
clnet, 706
CAI Unicenter management systems, 686
Calendar applet, 584
Centura, 685
CERN, 112
CGI and, 514
content, 441
CorelDRAW!, 720
Corel Photo-Paint, 718
Corel Xara, 722
Crunch!, 705
Cyberdog, 963

IPR2024-00143
Apple EX1011 Page 534

CyberMad, 486, 499-509
animation, 508
cover page, 508
interface design, 499-503
navigation, 507-508
page layout, 503-507
Cygnus, 773
databases
Access 97, 682-683
CAI, 686-687
Centura, 685-686
CGI, 680
IBM DB/2, 685
Informix, 687
Lotus Approach 97, 681-682
Oracle, 684-685
problems, 680
SQL Server, 684
Sybase, 683-684
defined, 859
design, connections, 114
Dynamic HTML, 415
Equilibrium, 723
Excite, 481
Fractal Design Corporation, 444
Fractal Expression, 720
Fractal Painter, 716
Galahad, 963
Gamelan, 562, 589
GIF Construction Set, 724
GIFBuilder, 256
GifBuilder, 724
GIMP, 717
glyph servers, 810
HomeSite, 964
HotDog, 965
HoTMetaL Pro, 965
HTML Guru, 962
HTML table model, 943
HTML Validator, 700
HTML Writers Guild, 962
HTTP, 943
ICE database, 686
InContext WebAnalyzer, 701
Informix, 687
iWORLD, 772
J-Track applet, 587
JARS, 588
JASC, Inc, 715
Jasmine database, 686
JavaScript documentation, 546
JavaSoft, 255, 588
JDK download site, 250
Lexical Analyzer for HTML and Basic SGML document, 943
LiveImage, 274
look and feel of site, 441-442
Lotus Approach 97, 681
Lycos, 479
Lynx, 963
M. Casco, 833
Macromedia, 666
Director, 725
Flash, 725
FreeHand, 721
ImageMap PowerApplet, 585
xRes, 718
Map This!, 273
MapEdit, 272
Microsoft
FrontPad, 965
FrontPage 97, 965
GIF Animator, 724
Image Composer, 718
Internet Explorer, 963
Site Builder Network, 960
Sitebuilder, 437
Missoula Rural Fire District, 292
guest book, 313-315
home page, 308-311
nested list page, 311-312
template, 301-308
MultiGIF, 725
navigation, 473-478
ActiveX controls, 477
CSS, 475
frames, 476-477
image links, 475
image maps, 475
Java, 477
JavaScript, 477
links, 474
search engines, 481-483
site maps, 478-481
visual cues, 477-478
NCSA Mosaic, 964
Net.Data, 685
Netcraft, 771
NetImpact Dynamo, 683
Netscape, 414, 562, 696, 962
Composer, 965
DevEdge OnLine, 960
Novartis, 440
O'Reilly and Associates, 775
OpenType fonts, 810
Opera, 964
PageMill, 966
PC Magazine Online, 706
PC Resources for Photoshop, 714
Personal Web Server, 739
Philips/Magnavox, 132
Photoshop online reference guide, 715
Photoshop Paradise, 715
planning for future technology, 836-837
plug-in sources
3D plug-ins, 663-665
animation plug-ins, 663-665
audio plug-ins, 659-660
business plug-ins, 676-677
conferencing plug-ins, 672
image plug-ins, 675-676
multimedia plug-ins, 663-665
video plug-ins, 655
POV-Ray, 727
Ray Dream Designer, 727
RealVideo, 726
Sams.net, 706
San Diego Supercomputer Center, 632
Sausage Software, 698

Silicon Graphics Inc, 632
site goals, 439-440
Novartis site, 440
SlipKnot, 964
Sony, 132
SPG, 717
SQL Server, 684
Standard Default Color Space document, 943
Sun Microsystems, 776
Sybase, 683
themes, 439
Ticker applet, 586
Travelution, 486-498
interface design, 488-492
navigation, 494-496
page layout, 492-494
Planetcam, 497
Tucows, 706
Ultimate Compendium of Photoshop Sites, 714
Unicode Consortium, 797
VivoActive, 726
Voodoo JavaScript Tutorial, 564
VRML
Architecture Group (VAG), 633
cafe, 632
models, 632-633
W3C, 65, 320, 943, 946, 960-961
Cougar, 961
CSS, 961
Dynamic HTML, 961
HTML 2 specification, 961
HTML 3.2 specification, 961
URL descriptions, 961
WBL (Web Baseball League) example, 417
data source, wblteam.txt file listing, 430
data-aware pages, 423-430
dynamic graphics, 420-423
dynamic text, 417-420
features, 417
Web Design Group, 962
Webber, 965
WebTV, 133
WebTV Network, 132
whirlGIF, 256
Xfractint, 728
XML specifications, 783
Xpire Plus, 703
Yahoo!, 961
web.sql program, 683
Webber Web site, 965
WebSite Pro Web server, 775
WebSite Web server, 775
cost, 775
O'Reilly and Associates Web site, 775
technical support, 775
WebSTAR Web server, 776
cost, 776
trial copies, 776
WebTV, 132-133, 859
background colors, 133
designing pages for, 132
Web site, 133
WebTV Network site, 132
WebTV Network Web site, 132
well-formed documents, 859
well-formed XML documents, 784-785
What you see is what you get (WYSIWYG), 860
whirlGIF Web site, 256
white space, 240
JavaScript style sheets, 409
white-space attribute, 956
whiteSpace property, 409
wide area information service (WAIS), 859
width, borders, 954
width attribute
<HR> tag, 177
<IFRAME> tag, 330
<IMG> tag, 239
<LAYER> and <ILAYER> tags, 351
<MARQUEE> tag, 258
<OBJECT> tag, 334
<PRE> tag, 161
<TD> and <TH> tags, 208
text boxes, 955
width property, JavaScript style sheets, 408
Window object, 599
referencing, 601
Windows, 859
animated GIF files (creating), 256
Web servers, 770
Windows 95, 770
Windows NT, 770
windows, defined, 859
Winfract, 727
WIRL plug-in, 667
wizards
Explorer, 733
Corporate Presence, 733
Discussion Web, 733
Import Web, 733
FrontPage Editor, 736-737
Database Connector, 736
Form Page, 736
Frames, 737
Personal Home Page, 737
Link Wizard, 745
Script Wizard, 610-613
Web Publishing, 740-741
connection information, 741
providers supported, 740
word processing, physical styles, 163
word wrap
disabling, 162
table alignment, 199
word-spacing attribute, 951
words, defining, 167
World Wide Web Consortium, see W3C
World Wide Web, see WWW

IPR2024-00143
Apple EX1011 Page 535

Write method, 601
.wrz file extension, 632
WWW (World Wide Web)
defined, 859
history, 13-14, 19-21
Berners-Lee, Tim, 13-14
see also browsers
WWW-Authenticate header, 534
WYSIWYG (What you see is what you get), 860
editors, 965-966
Backstage Internet Studio, 966
Microsoft FrontPad, 965
Microsoft FrontPage 97, 965
Netscape Composer, 965
PageMill, 966

X

Xanadu, 11
Xfractint, 728
XML (Extensible Markup Language), 782-784, 844
advantages, 783
attributes, 785
documents
valid, 785-786
well-formed, 784-785
DTDs
accessing, 786-787
advantages, 785-786
element declarations, 787-788
entity declarations, 788
extended locators, 789
future of, 790
linking capabilities, 788-789
play.xml, 786
DOCTYPE statement, 786-787
XML declaration, 786
specifications, 783
syntax, 785
tags, 785
<XMP> tag, 163
Xpire Plus, 703-704
disadvantages, 704
New! icons, 704
Web site, 703

Y - Z

Yahoo! Web site, 961

zindex attribute, <LAYER> and <ILAYER> tags, 351
ZINDEX values, 420

IPR2024-00143
Apple EX1011 Page 536