

Welcome to the Web

Lesson 1

Building your first web page

How the Web works Page 1 of 6

Because you're taking this course and reading this lesson, it's a safe assumption that you can get on the Web and move around with a certain adeptness. You most likely know how to access any given Web page using its Web address -- technically called a **URL** (Uniform Resource Locator) -- and how to click on hyperlinks and graphics to move from one page to another. In general, the Web is a seamless world; you often move from one Web site to another and may not know you've done so. If you want to add your Web page -- and eventually your Web site -- to this virtually seamless environment, you'll need to understand a bit about the workings behind the Web scenes.

Of clients and servers

Simply put, the Web is a gigantic network. What that means in non-geek speak is that there are, in the most general terms, two roles that a computer can play on the Web: **client** and **server**. Servers hold documents, images, sound files, and anything else deliverable over the network, and clients access those files. Occasionally, a computer can play both roles, but more often than not, a computer is one or the other.

For example, when you type the URL `http://www.yahoo.com/home.html` into your Web browser, your browser is sending a request to the Yahoo Web server -- (identified as `http://www.yahoo.com/`) for a Web page called `home.html`. When the Web server receives your request for `home.html`, it looks to see if the Web page exists, and if it does, the server sends a copy to you so that you can view it in your browser. If the page doesn't exist, you receive an error message letting you know that the browser can't find the page you asked for. This system of requests and responses is how Web clients and Web servers communicate with one another.

The Web wouldn't work without clients and servers. Web documents are stored on servers all over the world, and clients can access each of these servers in the same way, regardless of where the client or server is actually located. This means you can request pages from servers in Sydney, Australia, and Van Horn, Texas in the exact same way and receive responses from each server in the exact same way. In the end, Web surfing is nothing more than a Web client -- a.k.a. a Web browser -- requesting a series of Web pages from Web servers located all around the world.

How computers communicate Page 2 of 6

Even though the world is made up of all kinds of computers running all kinds of operating systems -- PCs, Macs, and Unix systems, to name just three -- clients and servers can be any kind of computer running any kind of operating system. You may surf the Web with a Macintosh client, while the computer serving you a Web page is a PC running Windows. A Unix computer running Linux may serve the next Web page you view. The beauty of it all is that you, the user, don't know the difference. A Web page is a Web page, no matter what kind of computer it lives on.

Different kinds of computers don't usually play well together, and it takes a bit of jumping through hoops to make them communicate. So why is it that the Web -- which is made up of all kinds of computers -- works so seamlessly? It's simple, really. All of the clients and servers on the Web speak a common language called

The common theme of hypertext

Did you notice that the language for building Web pages, HTML, and the protocol for exchanging

the **HTTP protocol**.

Common ground

A **protocol** is a set of rules two computers use to communicate with one another. Web browsers and Web servers both speak a protocol called HTTP (Hypertext Transfer Protocol) that carefully defines how Web pages are requested and received. As long as both the browser and the server speak HTTP, it doesn't really matter which operating system either is running. The HTTP protocol is the common ground that allows them to communicate.

Web sites like Yahoo and Microsoft that receive hundreds of thousands of hits a day actually use a collection of Web servers to respond to client requests. The Web clients may not necessarily be running the same kind of operating system as the Web server they are requesting Web pages from. Protocols fill the communications gap between different kinds of computers and allow them to exchange Web pages simultaneously.

What this means to you

You may be wondering why you should care about what goes on behind the scenes of the Web. As a Web surfer, you don't need to care, and in fact the Web is designed so that you won't have to. As a Web builder, you need to know enough to find a place to serve your Web pages -- a Web server of some kind -- so that your users can request your pages using their Web browsers. The world can't see your Web pages unless they reside on a Web server hooked up to the Internet. Once your pages reside on a connected server, however, they'll be available to anyone with a Web client (remember, that's a fancy name for a browser) to access. For now, that's all you need to know about Web clients and servers. In the last lesson of the course, you'll learn more about finding a Web server home for your documents.

Where HTML fits in

We've talk about servers and clients and protocol but not much about what's behind the scenes of Web pages themselves. The actual Web pages that clients and servers use the HTTP protocol to request and send are written in HTML (Hypertext Markup Language). While you don't have to learn more about HTTP than is in this lesson to build your first Web page, you'll need to know quite a bit about HTML -- and that's what this course is designed to teach you.

them, HTTP, both begin with "HT?" The **HT** stands for hypertext, because that's what Web pages are: hypertext documents. Hypertext is simply text that you can navigate through in a non-linear way using navigation tools. You use the **Hypertext Markup Language** to build documents that you serve to Web browsers from Web servers using the **Hypertext Transfer Protocol**.

What it takes to build your little piece of cyberspace Page 3 of 6

Now that you know a bit more about the technology behind the Web, you can begin to stake your claim and build your little piece of cyberspace. You'll be surprised at how easy it really is.

Gather some tools

To build a Web page and make it available for the entire world to see, you'll need to gather a small but necessary collection of software tools that you will use to build your Web pages. The good

The course message board

You can join your fellow students and instructor for "class" on the course message board. Although we don't hold class at any particular time of day, you can post to the message board and read other message board posts any time of day. Be sure to visit the

news is that your tool options are many and varied on any operating system, so you can easily build a collection of Web-page-building tools that meets your specific needs.

The better news is that you can fill the toolbox to the brim with the tools you need without having to spend an arm and a leg. In fact, you may have some or all of the tools on your computer already. At a minimum, the tools you'll need to have on hand before you begin building your Web pages are:

- A text editor
- A Web browser

Eventually, you'll want to acquire these tools as well:

- A graphics editing tool
- An FTP client

To get started with this class, you only need a text editor and a Web browser. Before the class is over, you'll also need an FTP client, and you might want to pick up a graphics editor, but of the two, the FTP client is more important and much less expensive -- maybe even free.

You'll read more about gathering the right software on the next few lesson pages.

Build your page

After you gather these tools together, it's time to build your page(s). Later lessons show you how to:

- Gather the content that you'll be including on your Web page or pages. Content includes the text that users will read on your page as well as the graphics that will jazz it up and contribute to your user interface.
- Put your content together in a document or set of documents, using HTML. Next to information gathering, this part of the process requires the most work, and it's what the majority of lessons in this course focus on.
- Move your Web pages to a Web server connected to the Internet. Your options for hosting your Web pages are many. Lesson 8 shows you how to sort through these options and find the one that is best for you.
- Update and revise your site on a regular basis. Publishing information on the Web is much easier than publishing information in any other media, so users expect your site and its content to be fresh and up-to-date.

This collection of activities may look a bit daunting, but you'll find that after you've been through them once or twice they become as automatic as your morning shower routine -- and nearly as easy.

A caution against skipping steps

message board to post questions that you might have, or just to chat with your fellow students.

Business tools from HP

HP's collection of tools and information in the personal computing expertise center can help you choose the right hardware to build your web pages.



» [Personal computing expertise center](#)

The ads for a variety of software packages and Web hosting services promise that you can build and publish Web pages in 10 seconds or less, a rather different message than this course presents. The truth of the matter is that you can build and publish Web pages and even whole sites without following each one of these steps. Nevertheless, just because you can do something doesn't mean you should.

Each of the steps, from finding the right tools to keeping your site updated on a regular basis, addresses a fundamentally important issue in the development and maintenance of Web pages. If you start to skip steps here and there in the interest of time, you'll find that your immediate needs are met, but you will have created a set of problems that won't rear their ugly heads until much later.

The importance of learning HTML

You may be wondering why it's so important to learn HTML when there are tools that will do it all for you. The following reasons say it all:

- If you rely on a Web development tool to write your HTML for you, you only know as much about HTML as your tool does. When new features and versions of the HTML language come out, you won't be able to use them without upgrading your tool.
- If something on your page breaks and your tool can't fix it, you won't be able to fix it either. Sometimes tools do strange things and produce strange HTML code. If you don't know HTML, you won't be able to change the tool's output and fix your Web page.
- Web development tools often write HTML that works better in one browser than in another, or that looks great when you preview it in the development application, but not in a real-life browser. That's not good if you want to reach a wide audience. In most cases, you want your HTML to be browser-agnostic, which means that you want to use standard HTML.
- HTML is really, really, really easy to learn. You're not learning a programming language, you're learning a markup language, and the two are very different. HTML is intuitive and you'll find that it makes sense almost immediately. As a bonus, what you learn about HTML will help you learn other markup languages -- like XML -- later on.

A good text editor Page 4 of 6

On the previous lesson page, you learned that you need a couple of software tools to get started building your first page: a text editor and a Web browser. Because you will do all of your Web page creation in a text editor it is a very important tool in your Web page arsenal and it pays to know a bit about your options so you can find the editor that works best for you.

Unzipping archive files

EditPad Classic -- and just about every other Windows download on the Web -- is saved as

Basic text editors

A text editor does exactly what it sounds like it does -- it edits text. Believe it or not, an HTML page is nothing but text, which is why it can be served and displayed by any kind of computer. In the world of computing, text is the universal language. Because Web pages are made up of nothing but text, you can use any tool that edits text to build Web pages. If you're thinking that this means you can just pop open Notepad on your PC or SimpleText on your Mac and begin writing HTML, you are correct. In fact, those of us who were building HTML pages when the Web was new had to use these tools because the now-common Web publishing tools like FrontPage and Dreamweaver weren't even a spark in a forward-thinking software engineer's brain.

If you're working on a Windows system or a Mac, you already have the text editor you'll need for this course. Because Notepad and SimpleText are installed by default on all Windows and Macintosh systems, respectively, you're ready to go. The following discussion of specialized text editors is intended to let you know what your options are. Don't feel like you need to run out -- virtually speaking, of course -- and download a whole new software product. You can stick with your system's default text editor or try one of the ones discussed here; the choice is yours.

While you can use Notepad or SimpleText to build Web pages -- and there are Web developers who do -- other more feature-rich text editors are available, many with Web-specific tools included. There are also a number of Web page development applications that shield you from having to write HTML code with a WYSIWYG (What You See Is What You Get) interface. But, as I've said before, they aren't necessarily the best tools for those new to the Web world.

The two text editors of choice for this class are:

- For Windows users: EditPad Lite, written by Jan Goyvaerts. This editor is almost free -- the creator asks that you send him a postcard if you like it, a small price to pay for a great text editor. This editor replaces your standard Windows Notepad, but that's a good thing. You can read more about this editor and download a copy at <http://www.editpadpro.com/editpadlite.html>.
- For Mac users: BBEdit Lite, is a stripped-down version of the most popular Macintosh editor, BBEdit, a great beginning tool for Mac users, and of course it's free. Download a copy at http://www.barebones.com/products/bbedit_lite.html.

a .zip archive to make the file smaller and faster to download. You need WinZip to unzip the file and install EditPad on your system. Many Windows systems come with WinZip already installed, but if you don't have it you can get a copy from the WinZip site at <http://www.winzip.com>.

Macintosh files are either Binhex or Macbinary files, and tools for opening those files are built into MacOS 9 and later. If you're a Mac user you should be able to open up a Mac archive by double-clicking on it.

You can use anything that generates plain text to write HTML. That really does mean you can start building Web pages without spending a dime. However, the specialized software packages designed specifically to help you build Web pages do have their benefits. Most have sets of tools that creating page quick and easy, HTML-aware spell checkers that find your spelling errors but skip any code on the page, and drag-and-drop interfaces that let you quickly build hyperlinks and images. Two of the best HTML tools out there are:

- HomeSite (for Windows users) from Macromedia, available for download at <http://www.macromedia.com/software/homesite/>.
- BBEdit (for Macintosh users) from BareBones, available at <http://www.barebones.com/products/bbedit.html>.

These two tools take you to the next level of Web page development, and once you learn to use them really make it much easier to build Web pages. If you're feeling adventuresome I recommend you download one of these and try it during the course, but keep in mind that you don't need either to build your first Web page. HomeSite and BBEdit both have 30-day trial versions and cost right around \$100 if you decide to keep them. To give you a taste of how useful these tools can be, all of the screen shots in this course use HomeSite.

In the end, you really can't build Web pages without a text editor. Even if you become a world-famous Web developer, you'll always have your trusty text editor lurking on your system for quick page fixes and development.

Can I use Microsoft Word as my text editor?

Those who are new to HTML always want to know if they can use Microsoft Word or any other word-processing tool to build their HTML pages. Since word processors can generate plain text documents, the technical answer is yes. The more practical answer, however, is that you shouldn't.

Word processors store all kinds of unseen information in files, which can make Web browsers behave very strangely. When you try to save a document in a Word processor as an HTML file, the processor may add additional HTML to the document that you didn't expect. All of this is also true of WordPad. In general, it's best to stay away from word-processing software and use a plain text editor like EditPad, Notepad or SimpleText instead.

A great source for text editors

While the editors discussed in this lesson are my favorite four editors, they only represent a fraction of the text editors available for you to choose from. To find a smorgasbord of text editors visit TUCOWS at <http://www.tucows.com> and type "HTML editor" or "text editor" in the search window. To go directly to a list of text editors for your operating system do the following:

1. Click on the link to your operating system (Windows, Macintosh, or Linux) in the Desktop Software section of the home page
2. Click the Internet tab at the top of the page
3. Click the HTML Editors or Text Editors link in the Web Building Tools category

For the curious, TUCOWS stands for "The Ultimate Collections of Winsock Software," and it is the best online repository of software for all things Internet and Web-related. "Winsock" is short for "Windows socket," but don't let the name fool you. There's plenty of Mac and Linux software on TUCOWS along with the Windows stuff.

Just as you need a Web browser to surf the Web, you need one -- actually, more than one -- to build Web pages. Later, you'll learn to check your HTML code in a browser as you go to make sure that your page will really be displayed the way you expect it to be.

Don't forget the computer

Before you begin building your web domain, be sure you have

Because Web surfers use a variety of browsers to visit sites, we recommend that you view your HTML in several different browsers before you post it for the world to see. Web browsers are funny creatures, and sometimes they interpret the same HTML in different ways. While you can't hope to emulate every possible scenario in which your pages will be viewed, it's always a good idea to see how the different browsers interpret your HTML. This means you'll have several browsers installed on your computer at any given time. At the very least you should have the latest versions of Microsoft Internet Explorer (<http://www.microsoft.com>) and Netscape Navigator (<http://www.netscape.com>).

If possible, it's also a good thing to have the next-to-most-recent version of these mainstream browsers. Remember that it takes awhile before substantial numbers of people download and install new Web browsers. You might also want to install these other Web browsers to get an idea of how browsers outside of the mainstream display your HTML:

- **Opera:** <http://www.operasoftware.com>
- **Amaya:** <http://www.w3.org/Amaya>
- **Firefox:** <http://www.mozilla.com/firefox/>

Moving on

In this lesson you've learned a bit about how the Web works behind the scenes and found out how to equip your computer with the software you need to get started building Web pages. The next lesson delves right into how HTML works so you can build your first Web page. Before you move on to the next lesson be sure to complete the assignment for this lesson and take the quiz. Also, don't forget to stop by the course message board to see what your instructor and fellow students have to say about Web page building.

your work.



» **HP Compaq dc5100
microtower desktop PC**



» **Desktop buying guide**