

# **Creating Your Own Web Site**

**Prepared by  
Anthony Atkielski**

**14 May 2006**

**© 2006 Anthony Atkielski. All rights reserved.**

**URL: <http://www.atkielski.com/PDF/data/WebSite.pdf>**

No part of this document may be reproduced, transmitted, or stored in an information retrieval system, by any means or in any form whatsoever, without the explicit prior written permission of the author.

Trademarks are mentioned within this document for purposes of information only and remain the property of their respective owners. Their appearance herein is not intended and should not be construed as endorsement by or aYliation with the owners of the trademarks.

While the author has made his best effort to ensure the accuracy and timeliness of information in this guide, he cannot be responsible for accidental errors or omissions.

No part of this guide should be construed as legal advice.

# Table of Contents

<b>1 Introduction</b>	<b>1</b>
<b>2 Why the Web?</b>	<b>1</b>
2.1 What is the Web?	1
2.1.1 Worldwide Access to the Web	1
2.1.2 What a Web Site Provides	2
2.2 History of the Web	2
2.3 The Web Today	3
<b>3 Overview of the Internet</b>	<b>3</b>
3.1 General Principles	3
3.2 History of the Internet	3
3.3 The Internet Today	4
3.4 Operating Principles	4
3.4.1 Clients and Servers	4
3.4.2 IP Addressing	5
3.4.2.1 Static and Dynamic IP Addresses	6
3.4.3 IP Protocols	6
3.4.4 Ports and Sockets	6
3.4.5 The Domain Name System	7
3.4.5.1 DNS Computer Names	7
3.4.5.2 DNS Queries and Authoritative Nameservers	8
3.4.6 DNS Domain Administration	9
<b>4 Overview of the Web</b>	<b>9</b>
4.1 Basic HTTP Protocol	9
4.2 Web Page Content and HTML	10
4.3 Advanced Web Features	11
4.3.1 Cascading Style Sheets	11
4.3.2 Server-Side Includes	11
4.3.3 Common Gateway Interface	11
4.3.4 Server-side scripting	12
4.3.5 Client-side Scripting	12
4.3.6 Active Controls and Plug-Ins	13
4.3.7 Cookies	13
4.3.8 Forms	13
4.3.9 Query Strings	13
4.3.10 Secure Sockets Layer (SSL)	14
4.3.11 Protected Pages and Sites	14

4.4	Browsers and Page Rendering	14
4.5	URLs	15
4.6	Web Servers	15
4.6.1	Web Server Hardware	15
4.6.2	Web Server Software	16
<b>5</b>	<b>Building Your Web Site</b>	<b>16</b>
5.1	Design Considerations	16
5.1.1	Visitor Paths to Your Site	16
5.1.2	Navigation	17
5.1.3	Graphics and Color	17
5.1.4	Dynamic Content	18
5.1.5	Accessibility	18
5.1.6	Download Time	18
5.1.7	Handwritten HTML vs. Authoring Tools	19
5.2	Hardware Requirements	19
5.2.1	Client Computer	19
5.2.2	Internet Connection	19
5.2.2.1	Dial-Up Connections	19
5.2.2.2	ADSL Connections	20
5.2.2.3	Cable and Other Broadband Connections	20
5.2.2.4	Internal LANs	20
5.2.2.5	Wi-Fi Hotspots	20
5.2.3	Miscellaneous Hardware	20
5.3	Software Requirements	20
5.3.1	Text Editors	20
5.3.2	FTP Clients	21
5.3.3	Browsers	21
5.3.4	Image Editors	21
5.3.5	Web-Authoring Tools	22
5.3.6	Dynamic and Active Content Considerations	22
5.3.7	Operating-System Considerations	22
<b>6</b>	<b>Publishing Your Web Site</b>	<b>23</b>
6.1	External Hosting Options	23
6.1.1	ISP Hosting	23
6.1.2	Free Hosting	24
6.1.3	Web-Hosting Companies	24
6.1.3.1	Shared Virtual Hosts	24
6.1.3.2	Dedicated Virtual Host	24
6.1.3.3	Virtual Server	25
6.1.3.4	Dedicated Server	25
6.2	Hardware Requirements	25
6.2.1	Server Computer	25
6.2.2	Internet Connection	26
6.3	Software requirements	26
6.3.1	Operating Systems	26

6.3.2	Web server programs	26
6.3.1	Supporting Software	27
6.3.2	Other Functions of the Server	27
6.4	Security	27
6.4.1	Firewalls, Routers, and NAT	27
6.4.2	Viruses, Trojans, and Other Malware	28
6.4.3	Backups	28
<b>7</b>	<b>Promoting Your Web Site</b>	<b>29</b>
7.1	Out-of-Band Advertising and Promotion	29
7.2	General Linking	29
7.3	Search Engines	29
7.3.1	Placing Your Site in a Search Engine	29
7.3.2	Optimizing Your Placement	30
7.4	Rating Systems and Filters	30
7.5	The Need for TraYc	31
<b>8</b>	<b>Threats to Your Site</b>	<b>31</b>
8.1	Copyright Infringement	31
8.1.1	Infringements on Your Copyrights	31
8.1.2	Registering Your Copyrights	32
8.1.3	Respecting Other Copyrights	32
8.2	Trademark Infringement	32
8.3	Libel	32
8.4	Prohibited Content	33
8.5	Attacks Against Your Site	33
8.5.1	Viruses	33
8.5.2	Worms	33
8.5.3	Trojan Horses	34
8.5.4	Denial-of-Service (DoS) Attacks	34
	<b>HTML Quick Reference</b>	<b>35</b>



# Creating Your Own Web Site

## 1 Introduction

This paper explains how to create and publish a Web site. It covers certain basic and essential technical concepts concerning both the Internet generally and the Web specifically, and describes methods and considerations for designing, building, publishing, hosting, and promoting Web sites.

This paper was originally written as supporting material for a classroom course. The target audience is non-specialists with a desire to build and publish their own Web sites, who have a fair level of experience with desktop computers and use of the Web, familiarity with some common desktop applications (such as Windows Notepad), and at least a vague understanding of the Internet and its most popular applications (the Web itself, electronic mail, and so on).

## 2 Why the Web?

Presumably anyone reading this paper has already more or less made the decision to build her own Web site, but it won't hurt to review some of the reasons why a person might want a Web site, as well as some of the purposes that can be served by a Web site.

### 2.1 What is the Web?

The Web is the world's largest and most accessible repository of accumulated knowledge. It contains over six hundred billion pages of information, spread over roughly one hundred million individual Web sites, all of which are instantly accessible from any computer with a connection to the Internet, anywhere in the world.

These facts are quite impressive in themselves ... but they are all the more so when one considers that the Web did not even exist before 1990.

#### 2.1.1 Worldwide Access to the Web

Although the Web is nominally worldwide in extent, the need for a computer and an Internet connection still represents a significant practical obstacle to surfing it in many parts of the world. Only about one in seven of the 6.5 billion or so inhabitants of our planet is in a position to surf the Web—but that still works out to about a billion Web surfers. Most of the surfers are concentrated in areas of the world with well-developed telecommunications infrastructures, such as North America, Europe, and a significant minority of other individual countries (*e.g.*, Japan and Australia, to name just two). Sometimes getting access to the Web is more of a political problem than a technological problem: the wide-open character of the Web at the present time makes some political regimes very nervous, and so many countries still formally restrict Web and Internet access to a privileged few. Even supposedly free democracies attempt to censor the Web or restrict Web publication or access in some cases.

As a resident of one of the regions of the world in which Web access can almost be taken for granted, you can easily surf freely when and where you wish. Better still, you can *publish* on the Web if you so desire; and in so doing, you can instantly make your creations visible and accessible to a billion other people in virtually every corner of the world. Even a modest Web site may receive thousands of visitors a month from a hundred or more different countries, if its contents are interesting. Indeed, if you publish a Web site of your own, it's arguable that your Web site will have more influence on the world than anything else you might do in your life (unless you become a captain of industry or a movie star).

The magnitude of the exposure provided by a Web site should not be underestimated. It's a sobering experience when you receive your first e-mail from a real human visitor in a faraway place, and you realize that he typed it to you while sitting at a PC next to a rain forest fifteen thousand kilometres away from your home. And when you check the logs on your site and see people visiting from Kuwait, Ghana, Micronesia, Eritrea, and Togo, these places begin to seem a lot more real than they ever did in geography class.

### 2.1.2 What a Web Site Provides

A Web site is like a shop window, but with the world passing in front of it, instead of just the residents of your local neighborhood. A Web site is a way to spread the word—whatever that word might be. If you’ve always wanted a soapbox for your personal rants, a Web site can provide one; and no matter what opinions you might hold, you can be sure that both supporters and opponents of your opinion will visit your site in large numbers. If you have an obscure hobby, you might be surprised to find how many other people are interested in the same hobby if you publish a Web site about it. If you are a musician or writer, you can develop an audience for your work by making samples of it available on a Web site—and no matter what your style or level of talent, there will be people who love your work, and people who hate it.

If you want to share photos and thoughts with friends or family members in distant locations, a Web site can be a handy way to do so. You’ll want to protect your site with passwords or with other methods if you don’t want the entire world to see these photos and thoughts, however. The general rule is: Never make anything public on a Web site that you wouldn’t want to see on the front page of *The New York Times*. And always remember that anything released to the public Web cannot be recalled; even if you take down a Web site on which you’ve published something, you can be sure that copies of what you published will persist for decades somewhere out there on the Internet. So think twice before filling your site with nude photos of yourself or tales of your dog’s battle with intestinal parasites.

Finally, if you’re operating a small business, your site can be a shop window in a more literal sense, providing information about your products and services, your prices, your business philosophy, and contact information for your business. More advanced sites can also handle online ordering with credit-card payments, and if your products are in electronic form, you can even accept orders and deliver products in real time via your site. We won’t be going into detail on these so-called e-commerce sites in this beginner’s guide, but it’s useful to know what is possible in time.

Remember that Web sites can evolve: the first version of your Web site need not be the definitive one, and you can add to it, modify it, completely rebuild it, or eliminate it at any time. It’s often easiest to start out with a very small, modest site and gradually build upon it over time.

## 2.2 History of the Web

The Web is a relatively recent phenomenon, but the idea of a high-speed, cross-referenced repository of information has been with us for quite some time. A classic description of such a system was advanced by scientist Vannevar Bush in an article published in *The Atlantic Monthly* in 1945. He described a kind of information archive, which he called a *memex*, containing microfilmed pages of information with cross-references built into them, such that information on one page of microfilm could provide immediate links to other pages of microfilm containing related information—making it possible to jump instantly from one page to another in a non-linear way. He suggested that ready-made information repositories could be simply plugged into the memex, complete with extensive cross-referencing, ready for use. Although Bush was not well informed on the state of information science, his article was widely read, and his idea of linking information in a non-linear way, in particular, inspired many. His memex was missing the idea of a direct connection to a network of other memexes, which would have been the rough equivalent of the Internet; but then again, he wrote the article decades before the Internet existed, and he was still ahead of his time.

Years later, in 1963, Ted Nelson coined the term *hypertext* while teaching sociology at Vassar College. He used it to describe a system of reference materials containing embedded cross references—similar to the memex of Bush—such that one could move freely in a non-linear fashion from one page or document in the system to another by simply selecting a cross reference in the body of the page or document. This process was called *hyperlinking*, and today’s Web implements it in the form of the links on a Web page, which allow a surfer to move from one Web page to another with a simple click of the mouse.

A number of working hypertext systems were developed in the years that followed Nelson’s first use of the term. All of them were specialized systems used in very specific environments, *e.g.*, HES (used by NASA on IBM mainframe computers), FRESS, ZOG, PERQ, the Aspen Movie Map, and others. However, these early systems generally depended on a single repository of information collected and housed by a single organization in a single geographical location. The Internet already existed at the time, but it had not yet been combined with the hypertext concept.

Around 1990, Tim Berners-Lee at CERN in Switzerland worked with Robert Cailliau to develop a proposal for an idea that combined hypertext with a computer network. Berners-Lee called it the *World Wide Web*, and the name stuck, although it is shortened today to simply the *Web*. The WWW allowed researchers at CERN to build a database of documents that could be easily cross-referenced and searched, and made this database available on the Internet. Berners-Lee also wrote the first Web browser and the first Web pages.

Years passed and many changes and improvements and developments came about; we shall not explore all the details here. Suffice it to say that the Web developed into a “killer app” that drove explosive growth in the Internet and made the network a household commodity instead of an academic curiosity (electronic mail had existed long

before the Web, but for whatever reasons it never engendered the kind of growth that the Web did). The Web made the average person on the street aware of the existence of the Internet, and also motivated many non-specialists (that is, people who had no previous interest in computers) to obtain computers and use them for surfing the Web.

## 2.3 The Web Today

With more than half a trillion pages of information online, the Web contains roughly a hundred pages for every man, woman, and child on our planet. That's about 34 million times more information than the entire *Encyclopædia Britannica* contains. If you spent your entire life visiting Web sites, spending only one minute at each site, and surfing eight hours a day and five days a week, you'd visit less than a tenth of them all—in fact, new sites are being added faster than you can visit them. It is thus no exaggeration to say that the modern Web contains more information than anyone can ever hope to learn or even examine. And with millions of pages going online each day, this situation isn't likely to change. But at least it's nice to know it's all out there for you to visit.

## 3 Overview of the Internet

Now that we've spent some time hyping the wonders of the Web, we need to look at how it all actually works. We'll begin by studying the *Internet*—the enormous, worldwide computer network upon which the Web is built. Learning something about how the Internet works is a tedious but necessary prerequisite to understanding the Web itself.

### 3.1 General Principles

A computer network is a way of physically connecting computers together: in practice, this boils down to wires and optical fibers, as well as “rules of engagement” that govern how computers are to talk to each other over those wires and fibers. Computer networks are very much like telephone networks, except that a computer network allows for communication between computers, whereas a telephone network allows for communication between people. There are many parallels between the two types of networks, and indeed they often share the same physical infrastructure.

There are many computer and telephone networks in the world. Most of the world's telephone networks are linked together to form a worldwide telephone network that allows anyone in the world to call anyone else in the world just by pressing a few buttons on a telephone. And most of the world's computer networks are also linked together to form a worldwide computer network that allows any computer in the world to contact any other computer instantaneously. The worldwide telephone network doesn't really have a name—we just call it the *worldwide telephone network*. The worldwide computer network, on the other hand, *does* have a name of its own: the *Internet*. This is purely a quirk of history, and doesn't mean that there is anything fundamentally more special about the Internet as compared to the telephone network, but it does make the worldwide computer network a lot easier to refer to.

The Internet is not one giant, monolithic black box, as the name might lead one to believe, but is in fact an organized confederation of many thousands of smaller computer networks, in the same way that the world's telephone system is an organized confederation of smaller telephone networks. All the computer networks that are part of the Internet follow the same technical specifications or rules, and it is the adherence to these rules combined with the physical interconnection of the networks that makes them part of “the Internet.” Any organization with a computer network that is willing to conform to the Internet protocols and interconnect with the rest of the Internet can become a part of the Internet. And any computer that knows how to “speak” the Internet protocols can be connected to the Internet via any of its constituent networks.

Nobody owns the Internet, just as nobody owns the world's telephone network. Nobody has complete control over the Internet, although individual owners of networks that are part of the Internet obviously have control over their own networks, and governments have some control over the parts of the Internet that fall within their geographical jurisdictions. In fact, the Internet works only because thousands of individual network operators have voluntarily agreed to cooperate to make it work. A few organizations exist to centralize certain Internet administrative and technical functions, but no law forces anyone to deal with them; operators work with them simply because such coordination allows the network to function more efficiently. It's all based on a set of longstanding gentlemen's agreements, and it has worked much better over time than most laws ever do. (Not surprisingly, many other instances of successful international cooperation work in much the same way, with gentlemen's agreements advantageously replacing binding legislation.)

### 3.2 History of the Internet

The history of the Internet is quite complex (like that of its cousin, the telephone network), and knowing this history in detail is not essential to our purpose, so we shall only summarize it very briefly here.

The Internet has been around much longer than one might think. The concept behind it is more than forty years old. It was first suggested at the Defense Advance Projects Research Agency (DARPA), a research organization funded

by the United States Department of Defense. The first actual physical implementation of what was to become the Internet was in fact called the ARPANET, and it officially came to life way back in 1969, with just one lonely computer connected to it.

The ARPANET was innovative in that it was a *packet-switching* network, instead of a *circuit-based* network like the telephone system. In other words, the ARPANET allowed computers to communicate by sending information to each other in separate, relatively independent chunks, called *packets*, instead of requiring them to establish continuous connections with each other as telephones must do. This was an extremely important advance, because it allowed the ARPANET to maintain communication integrity even if part of the network was damaged—as long as packets could still be routed from one computer to another. The telephone network, in contrast, was vulnerable any time a continuous path could not be established between two telephones. This feature of the ARPANET was important to the Department of Defense, because it wanted a nationwide computer network that would continue to operate even if part of it were destroyed by a nuclear attack. Fortunately, no nuclear attacks ever occurred; but the ARPANET's successor, the modern Internet, has proven the wisdom of the packet-switching concept by remaining in service during other disasters, such as hurricanes, earthquakes, and so on.

What started as the ARPANET grew and prospered for decades, absorbing other networks and technologies, and increasing in size from a handful of computers to tens of thousands. At the same time, ownership and operation of the network gradually passed from the hands of academic, government, and research organizations to those of private and for-profit organizations (such as telecommunications carriers).

In 1972, the first “killer” (*i.e.*, broadly popular) application for the network, electronic mail, was developed by Ray Tomlinson. At the same time, the idea of connecting together a large number of relatively heterogenous networks became popular and was dubbed “internetting.” This term eventually gave the Internet its name. And in 1983, a set of well-defined technical rules for operating networks together, called *Internet protocols*, was adopted by all the interested parties. In 1990, the original ARPANET was decommissioned, and the Internet became an essentially freestanding, global computer network.

In 1990, the creation of the World Wide Web increased the growth rate of the Internet by orders of magnitude, and the Web continues to be a major driving factor in Internet expansion today.

In 1981, there were about 200 computers on the Internet. In 1991, there were nearly 400,000. And by 2001 there were over one hundred million computers on the network.

### 3.3 The Internet Today

At the time of this writing, there are some 400 million computers connected to the Internet around the world, in more than 200 countries and territories, and these computers are used by around a billion human beings to access the network.

The modern Internet is jointly owned and operated by a wide array of private and public entities throughout the world, with telecommunications companies being particularly prominent. Some supervisory technical and administrative functions are still overseen by semi-governmental, non-profit entities, most of which are still located in the United States.

The general trend is towards continued, extremely rapid expansion of the Internet, in terms of the number of connected computers, geographical coverage, and the number of human beings with access to the network. Increasing commercialization, censorship, and government regulation of the Internet also appear to be distinct and durable trends, for better or for worse. The World Wide Web and electronic mail continue to be the leading applications of the Internet for the average non-specialist user, although the number of other applications for which the Internet is being used continues to expand.

### 3.4 Operating Principles

We now come to the point where we must discuss the details of exactly how information is exchanged between computers over all those wires and fibers on the Internet.

#### 3.4.1 Clients and Servers

Most interactions between computers on the Internet involve one computer asking the other for something, and the other computer responding to the request. The computer that initiates the conversation and makes the request is called a *client computer*; and the computer that answers and fulfills the request is called a *server computer*. The concept of client and server is important on the Internet, particularly for services such as the Web.

For example, when you surf to a Web site on your computer, your computer is the client, and another computer holding the content of the Web site you are visiting is the server. Typically, if you are using an ordinary desktop computer at home or office, your computer always acts as a client, and never acts as a server. In other words, your computer contacts other computers on the Internet to request various things, but no other computer on the Internet ever attempts to contact your computer.

The concept of clients and servers extends not only to computers, but to the programs running on the computers. For instance, the browser program you use to surf the Internet (such as Microsoft Internet Explorer or Firefox) is called a *client* program, because it *initiates* conversations with other computers on the Internet. The programs that run on computers such as Web servers are called *server* programs, because they “listen” for incoming requests and *respond* to them, making the computer behave as a server.

Because client computers only make requests of other computers, and never answer unsolicited queries themselves, it is not necessary that they be permanently connected to the Internet, nor is it necessary that their names or IP addresses (which we’ll explain shortly) be known to any computers other than the ones with which they initiate conversations. This means that an ordinary client computer can be disconnected from the Internet whenever Internet connectivity isn’t needed—and in fact this is how most desktop and laptop computers are set up. Additionally, client computers can be protected by software and hardware mechanisms that reject any attempt by another computer on the Internet to start an unsolicited conversation; such mechanisms are referred to as *firewalls*.

The situation is different for servers: since they must be constantly ready to answer queries at any moment, they must be permanently connected to the Internet, and they must run 24 hours a day, seven days a week (remember, the Internet is worldwide, and there will always be somebody, somewhere who wants to contact the server). Furthermore, it’s impossible to completely isolate a server behind a firewall, because the server must be prepared to answer at least certain types of unsolicited queries from other computers ... after all, that’s the definition of a server. This means that servers must be very carefully configured and highly secured in order to prevent them from being compromised by dangerous traffic from other computers operated by bad people.

### 3.4.2 IP Addressing

Every computer connected to the Internet has a unique identification number, called its *IP address*. An IP address is rather like a telephone number in a traditional telephone network. Computers on the Internet communicate by exchanging packets of data (that’s the definition of a *packet-switched* network like the Internet), and each packet contains the IP address of the computer that sent the packet and the IP address of the computer to which the packet is being sent. A packet is thus a bit like an electronic envelope, with the IP addresses being the destination and return addresses on the envelope. However, IP addresses themselves are more like telephone numbers than postal addresses.

Internally to the Internet (that is, between computers), IP addresses are numbers of thirty-two *binary digits* (or *bits*) in length, which look something like this:

```
0000101000000000000000000100101
```

There are 4,294,967,296 possible combinations of 32 bits, so that’s the maximum number of IP addresses that can be assigned, worldwide. It might seem like a lot, but quite a few of the available addresses are already in use, and more computers are joining the Internet all the time. Fortunately, by the time the current address scheme runs out of available addresses, new enhancements to IP addressing that are already in the works will have vastly expanded capacity.

Human beings understandably find 32-bit numbers awkward to use and remember, and so they usually write IP addresses in a so-called *dotted-quad* format. In this format, IP addresses always consist of four numbers separated by periods, each number being between 0 and 255 inclusive. The dotted-quad IP address is the rough Internet equivalent of an ordinary telephone number that we might jot down on a notepad. The binary IP address we gave as an example above would be written 10.0.0.25 in dotted-quad format, which is clearly a lot less bewildering to human beings.

Still, even when dotted-quad format is used, numeric IP addresses can be difficult for people to memorize, just as long-distance telephone numbers are difficult to retain. Fortunately, we are not obligated to memorize IP addresses, because the Internet provides a free service similar to the “Directory Assistance” or “Information” services available for telephones. This Internet service is called the Domain Name System. Unlike the telephone services, however, the DNS is completely automatic—you don’t have to look up an IP address yourself, because your computer will do it automatically for you, via the DNS, whenever you give it the name of a computer.

For example, when you surf the Web, you can type the name of a Web site directly into your browser. Web sites are housed on computers (as we shall see later), and these computers have IP addresses just as all other computers on the Internet do. But your computer will accept the name you type and automatically contact the Internet’s DNS to look up the corresponding IP address. This is done in a small fraction of a second and without any necessary action on your part, and so it is much more transparent and convenient than the use of Directory Assistance with the telephone system. You need only remember names of computers to surf the Web; you never have to know their IP addresses.

IP addresses are assigned by the Internet Corporation for Assigned Names and Numbers, a non-profit organization. Usually the ICANN delegates responsibility for assigning entire blocks of IP addresses to other organizations, which in turn may subdivide the address ranges and delegate authority further. In the case of your desktop comput-

er or your Web site, your IP addresses are managed and assigned by the same Internet service provider (ISP) that provides your connection to the Internet. You cannot choose your IP address, because it is usually a function of your geographic location and other network design and administration constraints.

### 3.4.2.1 Static and Dynamic IP Addresses

IP addresses are *static* if they are assigned once and never changed (or almost never changed); they are *dynamic* if they change regularly.

Server computers on the Internet are usually assigned static IP addresses. This is done because servers must be easy for other computers on the Internet to find, and this is only practical if the IP addresses of the servers do not often change. Additionally, servers are normally operating and connected to the Internet 24 hours a day, and seven days a week, so changing their IP addresses while they are online and running is awkward.

Client computers on the Internet are often assigned dynamic IP addresses. This is done because client computers typically are not connected to the Internet continuously, and also because, by assigning a client computer an IP address on the fly whenever it connects to the Internet, it is possible to reserve a small pool of addresses and use them to serve a large pool of computers (assuming that only a fraction of those computers are connected at any given instant). This is exactly how most Internet service providers operate, assigning IP addresses to customer computers as they connect, and then releasing those addresses to a pool of available addresses after the computers disconnect (the pool of addresses has been delegated to the ISP by “upstream” ISPs and ultimately by the ICANN). It isn’t necessary for a client computer to have a static IP address because other computers on the Internet normally do not try to contact it. It’s even a bit more secure to have a dynamic IP address, because it prevents hackers and other bad guys from noting the address of a vulnerable computer and then returning another day to attack it over the Internet.

### 3.4.3 IP Protocols

At the most elementary level, the information flowing over the Internet is nothing more than an endless sequence of packets, each containing a stream of ones and zeroes. Making sense of these packets requires organization, and organization requires technical standards. On the Internet, such standards are called *Internet protocols*. Although the Internet is actually a confederation of many independent networks, all of the operators of these networks have agreed to operate them in accordance with Internet protocols, in order to make intercommunication among them possible.

There are hundreds of different Internet protocols, each adapted to a specific application of the network. The World Wide Web, for example, uses a protocol called *Hypertext Transfer Protocol*, or HTTP. Electronic mail uses several protocols, including *Simple Mail Transfer Protocol* (SMTP) and *Post Office Protocol* (POP). Services such as instant messaging use still other protocols. All of these protocols serve different purposes, but they have common features that allow them to be used simultaneously on the Internet.

A computer attached to the Internet will “speak” the protocol that is required for whatever service it is using (HTTP if it is surfing the Web, POP or SMTP if it is sending or receiving e-mail, and so on). Computers often communicate with each other using several protocols at once. It all happens very quickly and automatically, of course, so ordinary human users don’t need to concern themselves with Internet protocols. However, a webmaster needs a basic understanding of Internet protocols in order to develop and manage her Web site.

### 3.4.4 Ports and Sockets

Most Internet Protocols have numbers assigned to them, called *ports*. The relationship between protocols and ports is not cast in concrete, and so often one hears the term *well-known ports* being used to designate the particular ports that are *almost* always used for specific protocols. For example, the HTTP protocol used to surf the Web is assigned the well-known port 80, but it is possible for two computers to use the HTTP protocol with a different port number (as long as they are both aware of the change). We’ll explain why this might be important later in this paper.

Common port numbers include 80 for the Web protocol, and 25 and 110 for the e-mail protocols SMTP and POP. Ports with numbers below 1024 are said to be *privileged*, meaning that they are assigned by a central authority and are only to be used by “trustworthy” programs on a computer. Port numbers at or above 1024 are freely available for any use. We’ll see how this can be significant later in this paper. The highest possible port number is 65535.

When one computer contacts another over the Internet, it specifies not only the IP address of the computer it wishes to contact, but also the port number of the service it wishes to use. For example, if one computer wishes to send e-mail to another computer, it will contact the destination computer by specifying its IP address and the port number 25 (which corresponds to the standard SMTP protocol used to send mail). The combination of the IP address and the port number is called a *socket*. When the destination computer sees a request for connection coming from another computer on the Internet with a port number of 25, it can assume that the other computer wants to send e-mail.

In any conversation between two computers on the Internet, both computers must use sockets. In other words, when one computer contacts another in order to visit a Web site, it specifies the IP address of the computer that holds the

Web site, and (normally) the well-known port number 80, which corresponds to the Web protocol; however, at the same time, it also provides its *own* IP address and a “reply-to” port number to the other computer. This latter port number is usually chosen at random by the originating computer, and it tells the destination computer which port to use when it sends a reply. It is never a well-known port; it always has a number above the privileged range 0-1023. So a computer might request a conversation with a Web site on port 80, and specify a port number of 4817 for the reply. The Web site computer will accept the incoming connection on port 80 (and thus will know that it is a request for a Web page), and will send its reply to the originating computer on that computer’s port 4817.

Well-known port numbers, like IP addresses, are assigned by the ICANN. However, whereas the ICANN sometimes delegates authority for assignment of certain blocks of IP addresses to other organizations, it does not do this for port numbers, because there are simply too few port numbers to permit it (particularly in the privileged range of port numbers, below 1024).

### 3.4.5 The Domain Name System

We’ve already seen that the Internet provides a kind of automatic Directory Assistance for converting computer names to IP addresses. This service is called the *Domain Name System*, or DNS.

The DNS operates using a large number of server computers on the Internet, called *nameservers*. Nameservers provide translation from names to IP addresses. When another computer on the Internet requires the IP address corresponding to a given computer name, it contacts the nearest DNS nameserver and requests it.

If you’ve been reading carefully, you may have noticed a slight problem, namely, that a computer needs to somehow already know the IP address of the nearest nameserver in order to use the DNS to look up IP addresses for other computers. There are two ways to resolve this problem, depending on whether the computer using the DNS is a client (like a desktop computer) or a server (like a Web site). An ordinary client computer in a home or office will simply be informed of the IP address of the nearest nameserver when it first connects to the network, via an automatic procedure independent of the Internet that occurs during the connection process (the computers at the other end of the connection, at the ISP’s headquarters, will provide the information). In the case of a server computer, the system administrator of the server must manually give the computer the correct IP address of the nearest nameserver when she configures the Internet connection for the system.

#### 3.4.5.1 DNS Computer Names

In the Domain Name System, computers are given names. The complete name of a computer is called its *fully-qualified domain name*, or FQDN. The FQDN consists of one or more individual names made up of letters and/or digits, connected together by periods. A typical FQDN is `www.cnn.com`. The first name in the FQDN (`www`) is called the *hostname*. The other names in the FQDN are called *domain names*. The last name in the FQDN (`com`) is called the *top-level domain name* (TLD), and the domain name that immediately precedes it (`cnn`) is called the *second-level domain name*.

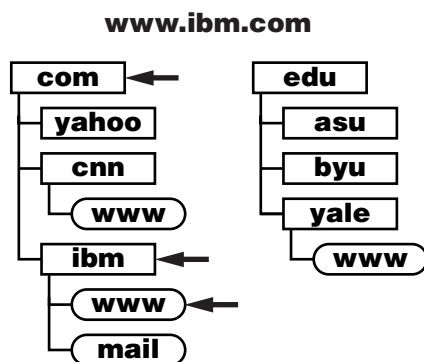


Figure 3-1. The DNS hierarchy

Computer names on the Internet are structured into a tree-like hierarchy (see Figure 3-1). At the top of the tree are the top-level domain names; there are relatively few of these, and most of them are familiar to all of us: `com`, `org`, `edu`, and so on. At the next level below that are the second-level domains, and there are many millions of these. Every second-level domain is associated with a top-level domain; second-level domains must be unique within their top-level domains, but other second-level domains can have the same name as long as they are subordinate to a different top-level domain. Thus, one can have `ibm.com` and `ibm.net` (the same second-level domain name in two different top-level domains), but one cannot have two second-level domains named `ibm` in the same TLD.

Within each second-level domain, there can be any number of unique third-level domain names. Indeed, beneath any level of domain, there can be any number of unique subdomains or hostnames. There is no fixed limit to the number of levels.

Servers usually have FQDNs that contain no more than three or four domain names, in order to make them easier to remember. A typical FQDN for a Web server might be `www.cnn.com`. A typical name for an e-mail server might be `mail.hotmail.com`. In this latter example, `mail` is the hostname, `hotmail` is the second-level domain, and `com` is the top-level domain.

In Figure 3-1, arrows indicate the domains making up the FQDN `www.ibm.com`, as an example. The actual computer hostnames are shown as ovals; the domains above them are shown as rectangles.

Every computer on the Internet must have an IP address, but it need not have a name. Client computers don’t always have names, since nobody needs to refer to them (they initiate requests to other computers, but they do not respond

to requests, by definition). If they do have names, they are often very long and complex, because they are usually assigned by network administrators who use the names to help identify the location of the client computer on the network, in order to ease maintenance and facilitate troubleshooting. A typical client FQDN might be something like `m44287-80.cax.phx.dl.east.bignetwork.net`.

The leftmost name in a fully-qualified domain name is the hostname, and identifies a specific computer. By convention, computers that are Web servers have the hostname `www`, which is why most Web site names begin with `www`. Computers that are servers for other purposes will often have names that make this clear, *e.g.*, a server that handles electronic mail will often have a hostname of `mail` (as in `mail.ibm.com`). Again, this is only a widespread convention, not a technical requirement.

The DNS allows computer to have more than one name. It has one “main” name, and can have any number of alternate names or *aliases*. For example, a computer that handles both Web traffic and e-mail might have two aliases, `www.xyz.com` and `mail.xyz.com`. Small businesses that have only one server on the Internet often use it for multiple purposes and give it multiple DNS aliases.

Conversely, a single FQDN can refer to several computers, in certain specific conditions. This possibility is exploited in very large organizations that have such a heavy load on their Web or mail servers that a single computer cannot handle it; in these cases, a single name, like `www.xyz.com`, may refer to more than one server, and a computer looking for the name will be given the IP address of one of the servers going by that name, with the exact server being chosen by the nameserver in a way that helps to balance the load.

### 3.3.5.2 DNS Queries and Authoritative Nameservers

No nameserver has information on every computer name and IP address on the Internet; that would require too much space and would be too difficult to keep up to date. Instead, every domain in the DNS has its own *authoritative nameserver*. The authoritative nameserver is a nameserver that contains the most up-to-date names and addresses for that particular domain—it is the final authority for information on that domain. A single nameserver may be authoritative for multiple domains, but a single domain has only one authoritative nameserver (actually two, but since one is just an identical back-up for the other, this is functionally the same thing).

Most nameservers maintain a list of the names and addresses that they’ve looked up recently, called a *name cache*. If a new query arrives that references a domain that a nameserver has already looked up, it will give the same response it gave before. If it doesn’t have information on that domain, or if too much time has elapsed since the last time it was looked up and the information might be “stale,” the nameserver will attempt to locate the authoritative nameserver for the requested domain and obtain a fresh copy of the name and IP address for that domain. In this way, most DNS queries are answered very quickly, but nameservers do not have to know about every single computer on the Internet; and when name and address information are updated, they propagate outward to the entire Internet in 24 hours or so, as the cached information in each nameserver goes stale.

A computer on the Internet will normally query its nearest nameserver in order to translate a computer name to a computer IP address. If the nameserver doesn’t have any information on the requested computer name, it will try to locate the authoritative nameserver for that domain. If it cannot locate an authoritative nameserver, it will pass the query “upstream” to another nameserver, and this latter nameserver will repeat the process. This continues until the domain is found. Ultimately, if no lower-level nameservers have the requested information, a query will end at a *root server*, which is a “master” nameserver that contains pointers to every top-level domain in the world. The query can then trickle back down in search of the domain. If this still fails, the original nameserver that received the query initially will eventually send an error message back to the computer that requested the IP address for the computer name.

DNS queries are answered in a small fraction of a second under good conditions. However, cheap ISPs that attempt to cut corners on equipment sometimes have badly overloaded DNS servers that may take up to several seconds or more to answer a query. When you see a message similar to `Looking up www.xyz.com` in the status bar of your browser while surfing the Web, this means that the browser is waiting for a reply to a DNS query for the computer name `www.xyz.com`.

There are just over a dozen root servers. Most of them are widely scattered geographically, so that the loss of one or two root servers would not have any effect on the Internet. The IP addresses of the root servers do not change, because the world’s nameservers must know how to locate them without the need to look up names or addresses. The Internet would still function even if all the root servers were destroyed; but the DNS would not work, which would make the Internet very difficult to use (everyone would have to specify numeric IP addresses instead of just typing computer names).

At the very top of the DNS hierarchy is an invisible domain, the *root domain*, which is above the top-level domains and has no name. Since every fully-qualified domain name ends with the nameless root domain, most of the time it is just ignored by human beings, and computers just quietly tack on the domain when they submit DNS queries. The root servers are the authoritative nameservers for the root domain.

### 3.4.6 DNS Domain Administration

Every domain on the Internet has an owner and manager. The owner of a domain decides which domains can or cannot be created subordinate to her (or its) domain, although the owner does not control domain levels below that. In other words, the owner of the edu domain can decide which domains are defined subordinate to edu, such as yale or harvard, but it cannot control the domains defined subordinate to yale.edu or harvard.edu; the owners of the respective subordinate domains control that.

The top-level domains are managed by different organizations. The most popular top-level domains are “generic” TLDs (or *gTLDs*) that are theoretically managed in the public interest for the entire world; these include the com, org, net, and edu TLDs, amongst others. The com and net gTLDs are managed by VeriSign, Inc.; the edu gTLD is managed by EDUCAUSE, and the org gTLD is managed by the Public Interest Registry.

A full set of top-level domains has been established corresponding to each of the individual countries in the world, and in fact the majority of the more than 260 top-level domains defined are “country-code” domains, or *ccTLDs*. By convention, ccTLDs have two-letter designations, instead of three. The ccTLD of each country is managed by an organization appointed or created by the country’s own government. For example, the ccTLD of the United Kingdom is uk, and the UK government has designated Nominet as the manager of the domain.

Some TLDs are legacies of the original US ownership of the Internet. For example, gov is reserved for US government entities, and mil is reserved for the US military. The former is managed by the US General Services Administration, and the latter is managed by the US DoD Information Center.

There exists a handful of other TLDs that have been created for various and sometimes dubious reasons, including pro, travel, biz, info, and others. None has enjoyed any real success. The overwhelming majority of computers in the world, particularly those with any type of business use, are in the com TLD.

The root domain is managed by the ICANN, the same organization that delegates management of IP address ranges and port numbers.

There are “alternative” root domains on the Internet. These are completely independent domain hierarchies with their own, separate root servers and DNS nameservers. They are only accessible to people who choose to use their nameservers, and they haven’t gained widespread acceptance, so we won’t discuss them further here. Remember, all computers on the Internet are on the same global metanetwork, but it’s possible to have multiple DNS naming hierarchies. Only the DNS hierarchy managed by ICANN is considered “official.”

## 4 Overview of the Web

Now that we have the fundamental principles of the Internet out of the way, we can begin to look at the fundamental principles of the World Wide Web, which is the Internet service that interests us specifically in this paper.

The Web is a worldwide, instant information resource, unlike any other in history. It provides immediate electronic access to more than six hundred billion pages of information, from almost any point on the planet. The creation of the Web has effected a sea change in the way the world gathers, manipulates, communicates, and analyzes information.

Most of us are consumers of the Web; but since you are reading this paper, you presumably wish to become a publisher of the Web as well, by building your own Web site. Publishing information on the Web is much more complex an undertaking than seeking it out, but it’s not outrageously difficult.

### 4.1 Basic HTTP Protocol

The HTTP protocol is a set of rules that govern how a computer surfs the Web. In general, when a client computer on the Internet requests a Web page, the following operations take place:

- The browser program on the client computer contacts its nearest DNS nameserver to obtain the IP address of the server computer hosting the Web site that the browser has been instructed (by its human master) to visit.
- The nameserver replies with the IP address of the server for the Web site.
- The browser program initiates a conversation with the server of the Web site directly, and asks for the Web page or file that it wishes to receive.
- A Web server program listening for Web queries on the server computer receives the client computer’s query, retrieves the contents of the requested Web page or file, and transmits those contents back over the Internet to the client computer.
- The browser receives the contents of the requested Web page or file and takes appropriate action. For a Web page, the browser interprets the contents of the page and displays them with appropriate formatting in the brows-

er window. For other types of files, the action depends on the type of file (more on this later).

- The browser closes the connection with the Web server.

In the original version of the HTTP protocol, only one Web page or file was requested with each connection; if multiple pages or files were required by the browser, it opened a new connection with the Web server for each page or file. The current version of the protocol allows a single connection to serve for the transfer of several pages or files, if needed.

Surfing the Web thus involves two computers (a client computer and a Web server computer) and two programs (a browser program on the client computer and a Web server program on the Web server).

## 4.2 Web Page Content and HTML

A Web page is basically nothing more than a plain text file. When a browser requests a Web page from a Web site, the latter retrieves the contents of a text file and sends it to the browser. The text file can be created and stored on the Web server computer just like any other text file.

Of course, when you actually view the average Web page on a screen, it contains a lot more than just plain text: it often contains images, decoration, formatting of the text, and so on. This is made possible by the use of *hypertext mark-up language*, or HTML. HTML is just a fancy name for some special keywords, called *HTML tags*, that are inserted into the text of a Web page by its author to cause a Web browser to take special action above and beyond the simple display of text. HTML tags are enclosed in angle brackets ('<' and '>'), and they give a browser additional instructions on how to format a Web page.

For example, typical tags are <b> and <u>, which mean “display all the text that follows in bold characters” and “underline all the text that follows,” respectively. When a browser sees these tags embedded in a block of text in a Web page, it removes the tags themselves and displays all the text that follows in bold characters or underlined characters, as appropriate. Suppose that a sentence in the original Web page contains the following raw text:

This is an example of <b>bold</b> and <u>underlined</u> characters.

A Web browser will display the sentence above as shown below when it actually places the text on the screen:

This is an example of **bold** and underlined characters.

When the browser sees the tags <b> and </b>, it interprets them as delimiters enclosing text that is to be displayed in bold characters. Similarly, it interprets the <u> and </u> as delimiters enclosing text that is to be underlined. With most HTML tags, the tag itself means “start here,” and the same tag with a slash in front of the tag name means “stop here.” Thus, <u> means “start underlining here,” and </u> means “stop underlining here.”

HTML tags are not limited to controlling the way text is formatted. There are also tags that tell the browser to insert images or other content into the page as it is displayed. The `img` tag is used to place an image on a Web page when it is displayed. It looks like this:

```

```

This tag tells the browser to insert the image contained in the file `/images/happyface.jpg` at the point where the tag occurs in the Web page. The tag itself is removed, as always, and the image is put in its place. The `src` attribute in the tag tells the browser where to find the image; the string of characters within the quotation marks is interpreted as a URL (more on this later) identifying the image file on the Web site from which the Web page was obtained. Thus, the browser must contact the Web server again using the HTTP protocol in order to request the file that contains the image. When it receives the file, it displays the image as part of the Web page display in the browser window.

Notice that the `img` tag has only one form; there are no separate “start” and “stop” forms (as there are for the formatting tags like <b> and </b>) because they don’t make sense in this case.

Another important tag is the *anchor* tag, <a> and </a>. The anchor tag turns whatever is between the starting and stopping forms of the tag into a hyperlink that can be clicked upon to reach a different Web page. Consider this example:

```
<a href="index.html">Click here for index.</a>
```

When this text is parsed by the Web browser, it will be displayed as follows:

[Click here for index.](#)

If you move the mouse cursor on your computer over this text as it appears in the browser window, it will be highlighted in some way (usually with a change in color, but this depends on the browser and on the style of the Web page) to indicate that it is a clickable link to another page. If you then click with the mouse while the cursor is over this highlighted text, the browser will request whatever page is pointed to by the anchor tag (in this case, it will request a page called `index.html` from the same Web site as the one that supplied the current page). Most browsers automatically underline text that is part of a link (as above), but this can be changed.

There are many other tags that can be used in a Web page; a list of the most common among them appears at the end of this paper. None of the tags are strictly necessary in order to get a page to display correctly on the screen, but there are standards and conventions about which tags should be used and when, which we also discuss briefly at the end of this paper.

### 4.3 Advanced Web Features

Some advanced features of the Web, while widely employed on the Web overall, are not usually used by webmasters who are building their first Web sites. These are mostly outside the scope of this paper, but we summarize some of the more prominent among them below just for the sake of completeness.

#### 4.3.1 Cascading Style Sheets

Most Web sites today use *cascading style sheets* (CSS). Cascading style sheets are a method of controlling various aspects of the way a Web site is presented without individually changing thousands of HTML tags in every page on the site.

For instance, the `<b>` tag normally causes text to appear in bold characters. However, with CSS, it is possible to redefine the tag so that it applies any desired formatting to characters. Thus, a webmaster could use CSS to adjust the `<b>` tag so that it turns characters red instead of, or in addition to, making them bold. Since style sheets can be stored in files separate from the actual Web pages that use them, a single change to a single CSS file can change the “look and feel” of an entire Web site, by modifying the effect that certain tags have. In the example just given, a webmaster could change all the bold text on a site so that it appears not only bold but also underlined or in green, and she could do this by making only one modification to one file on the site. Without CSS, every page on the site would have to be extensively modified to change the tags.

A CSS directive to change make all bold text red would look something like the following:

```
B {color: red}
```

This directive can be put directly within the page it affects, enclosed by the special style tags, or it can be placed in a separate file that can be referenced by every page that needs the style. Usually the latter method is used, so that the style of all the pages on a site can be changed with one modification to one CSS file.

CSS is not mandatory, but it becomes increasingly useful as a site grows to include more and more individual pages.

#### 4.3.2 Server-Side Includes

A *server-side include* (SSI) is a way of inserting text from one file into the text obtained from another file, when the latter file is transmitted as a Web page. A special directive is placed in the text of a Web page by the webmaster, and when the Web server sees this directive as it delivers the page, it automatically inserts text from another page in place of the directive. SSI is used to insert “boilerplate” text into Web pages, when many pages require the same text, such as, say, a copyright notice. A SSI typically looks like the following:

```
<!--#include virtual="/includes/counter.html" -->
```

This example will insert the contents of a file called `/includes/counter.html` into the current Web page file whenever the file is delivered to a browser. The browser will receive the text of both files exactly as if they were a single file, with the text of the inserted file replacing the SSI (which is removed by the server). In this particular example, the inserted text adds a page counter at the bottom of the Web page.

Like CSS, SSI allows a webmaster to save time and energy by making changes to only one file on the site, instead of changing every page individually.

#### 4.3.3 Common Gateway Interface

Most Web servers provide a method for executing programs on the server in place of delivering a text file in response to a Web request. In other words, a server can be configured such that any request for a certain page triggers the execution of a special program on the server, which then generates text output that is sent to the requesting browser in place of the requested page. This technique is called the *Common Gateway Interface*, or CGI.

CGI can be used to generate *dynamic* content, meaning Web pages that are different for each visitor or contain information that changes each time the page is requested. This is in contrast to the normal, *static* Web pages that are simply text files that the server sends to the requesting browser.

CGI programs are actual computer programs, and writing them requires programming skills. The usual purpose for using them is to generate a Web page dynamically based on information from a database or from input from the person visiting the Web site. The programs can be written in a variety of computer programming languages, the most common choices being the C or Perl languages.

A typical use of a CGI program might be to create a Web page that displays the contents of a database (this is very common on many commercial sites, especially online shopping sites such as Amazon.com). The CGI program reads

the database and generates a text file containing HTML tags that contains the results of the database search. The page is thus generated on the spot each time a computer asks for it, and it can be customized on each occasion with slightly different contents—the same “current account status” page might display different information for different visitors, for instance.

The webmaster of a site must configure the Web server to run CGI programs when certain Web page URLs are requested by visitors. Each time a visitor requests one of these Web pages from the site, the Web server runs the appropriate CGI program and directs its output to the requesting visitor’s computer, in place of the ordinary text file that would normally be sent to the visitor. Since the result in both cases is plain text with HTML tags, from the standpoint of the visitor’s client computer and browser, both operations produce perfectly legitimate Web pages that can be rendered normally.

#### 4.3.4 Server-side scripting

Server-side scripting is a simplified variation on the theme of CGI programs. Instead of a freestanding CGI program written in a conventional programming language, server-side scripting uses a “scripting” language that is inserted directly into the text of a Web page.

Scripting languages are like HTML tags, but much more elaborate in that they contain instructions to be executed instead of mere parameters. In server-side scripting, the scripting language instructions are read and interpreted by the Web server when it retrieves the text of the page, then the scripting language itself is stripped from the text of the page and the text it generates is put in its place before the page is sent to the requesting client computer.

For example, when the PHP scripting language is used in a server-side script, the following PHP instructions might appear in the text of the Web page when it is retrieved by the Web server:

```
<?PHP echo($HTTP_SERVER_VARS['REMOTE_ADDR']); ?>
```

These instructions tell the Web server to replace the instructions in the text with the IP address of the client computer accessing the Web site. When the Web server reads the text of the Web page in preparation for sending it to the client computer, it sees these instructions and executes them, effectively replacing them with the IP address of the visitor’s client computer before the text is sent to that computer. So the client computer receiving the text of the Web page will see

10.48.101.15

in the text of the Web page instead of the script instructions (assuming that 10.48.101.15 is the IP address of the client computer).

The advantage to CGI and server-side scripting is that the pages they generate dynamically look exactly like normal, static Web pages to Web browsers, so no modification of Web browsers is required. By the time the Web page reaches the browser, it *is* static. The visitor to the page doesn’t know that the page was created dynamically and was (possibly) customized specifically for him. Another advantage of these techniques is that they allow a Web page to be built dynamically based on information on the Web site (such as product or customer databases, etc.)—information that would not otherwise be available to site visitors.

The disadvantages of these techniques are that the content of the page cannot be changed once it is sent to the client computer (because at that point it has become just like any static Web page), and the content cannot be based on any information residing on the client computer, because the Web server has no access to that. A server-side script cannot insert the current time on the visitor’s PC into a Web page, for example, because it doesn’t know what the local time is on the visitor’s computer (but it *can* insert the local time on the Web site, of course).

#### 4.3.5 Client-side Scripting

Client-side scripting is similar to server-side scripting, except that it is implemented on the client computer, inside the Web browser. The scripts are contained inside the Web page, and they are transferred to the browser along with all the other content of the page. The browser then executes the script.

Like server-side scripting, client-side scripting is also a form of programming, but the most common language used is Javascript, rather than PHP or Visual Basic.

An advantage of client-side scripting is that it can change the contents of a page after the page has been sent to the browser—because the scripts are executed by the browser on the client computer, and not by the server. This is also a disadvantage, though, in that anything that must be done before the page is sent to the client computer (such as checking a database on the server) cannot be accomplished with client-side scripting.

Another disadvantage of client-side scripting is that it is not transparent to the browser. A Web server can execute server-side scripts without the browser ever knowing that such scripts have been used, but a client-side script requires that the browser on the client computer be able to execute the script. Most browsers today can execute scripts in Javascript, but older browsers (and some special-purpose browsers) cannot, and other scripting languages may not be supported. Additionally, some users turn off scripting in the browser for security reasons.

Both server-side scripting and client-side scripting present security issues. Server-side scripts are a potential security problem on Web servers; client-side scripts are a potential problem on client computers and browsers. Server-side security issues are usually not that serious, since the webmaster can control which scripts are executed by the server. Client-side security issues are much more serious, since the person surfing the Web has relatively little control over the client-side scripts that are executed when she visits a particular site (she can turn scripting off entirely, but that's not always practical, since client-side scripting is ubiquitous these days on the Web).

#### 4.3.6 Active Controls and Plug-Ins

Many browsers allow the use of *active controls* (such as Microsoft's ActiveX technology) or *plug-ins*. These are similar to client-side scripts, only much more elaborate. With these technologies, actual conventional computer programs are installed on the client computer and are executed under the control of Web pages.

Macromedia Flash is an example of an active control or plug-in. Flash animations are proprietary files that are downloaded as part of a Web page and are executed as programs on the local client computer, thanks to active controls or plug-ins that are installed directly on the client computer.

Because active controls and plug-ins are actual programs on the client computer, they can do anything that a normal program on the client computer could do, and this provides unlimited flexibility. Unfortunately, it also provides unlimited potential for malicious mischief, such as viruses, worms, Trojan horses, and the like. In many cases, the risk dramatically outweighs the benefit, and so many users turn these features off in their browsers for security reasons. Being able to watch a Flash cartoon is often not worth it if it means opening the computer to infection by viruses.

#### 4.3.7 Cookies

Cookies are small text files that are stored on a client computer at the request of a Web server. Once the file is stored, its contents are sent back to the same Web server each time the computer visits the corresponding site.

The purpose of cookies is to retain a small amount of information concerning a visitor to a Web site from one visit to the next, typically in order to memorize some sort of personalization of the site. For example, a Web site might use a cookie to store the name of a visitor, so that he can be addressed by name the next time he visits the site. The Web server decides on the content of the cookie and sends this content to the browser, which stores it in a local text file on the client computer for the specific site being hosted by the server. On the next visit to that same Web site (if any), the browser will send the contents of the cookie back to the server, so that it can reestablish whatever personalization it has in mind for the visitor.

Cookies have often been portrayed as serious security breaches, but in fact they are among the most innocuous of all advanced Web features. Their usefulness usually greatly outweighs the very small security risk, and so most Web surfers allow their browsers to store cookies (all the leading browsers allow users to disable cookies if they so desire).

The use of cookies requires server-side and/or client-side scripting and/or CGI programming, since some sort of programming is required to decide what to store on the client computer and what to do with the stored information when it is returned to the Web server on subsequent visits.

#### 4.3.8 Forms

Although Web sites normally display information with very little interaction from the site visitor (apart from clicking on links), it's possible for a Web site to request/allow considerable interaction through the use of *forms*. A Web form is a Web page that has open areas into which the Web surfer can enter information, which is then sent back to the Web server.

Web forms are used for things like site feedback, guest books, requests for technical support, database searches, and many other purposes. A typical example is the field on a search engine into which one types the words for which one wishes to search. The Web page that provides this field is a Web form. When an appropriate button is clicked on the page, the contents of the form as entered by the user are sent back to the Web server for processing.

Processing of Web forms requires extensive server-side scripting or CGI programming. For some common purposes, such as guest books and feedback forms, there are canned scripts available on the Internet that a webmaster can simply install on her own site, without too much need for programming expertise. But for most other purposes, special scripts or programs must be written to process the information returned from a Web form by visitors to the site, and this is not a trivial undertaking.

#### 4.3.9 Query Strings

*Query strings* are similar to forms. However, a query string is just a string of characters added to the URL of a Web page, and preceded by a question mark. In the example below, the query string is shown in bold:

<http://www.xyz.com/index.html?showfile=fred>

When a Web page is invoked with a URL that includes a query string, the query string is available to any scripts or CGI programs invoked for the page. Query strings are used to pass a few brief parameters to Web pages.

Like forms, query strings must be processed by scripts or CGI programs. The length of a query string is limited by the length of the URL that a browser can send and a server can accept, and query strings are best for brief, optional parameters that must be passed to a script on the server. Forms are suitable for larger volumes of data, when the data cannot fit on a single line after a URL.

#### 4.3.10 Secure Sockets Layer (SSL)

The *Secure Sockets Layer*, or SSL, is an Internet protocol that provides confidentiality and authentication for communications over the Internet. Its relevance to the Web resides in the fact that many e-commerce, financial, and banking sites use SSL to ensure that communications between the client computer and the Web server are kept fully confidential. SSL uses modern computerized cryptography to provide its confidentiality and authentication functions.

SSL also makes it possible to verify the identity of Web sites and client computers. SSL is routinely used for to keep communications between client and server computers confidential, and it is also commonly used to verify the identity of Web servers (so that surfers know that they are visiting a real site and not an imposter), but the client authentication aspect of the protocol, which allows Web servers to verify the identity of visitors, is rarely implemented.

Using SSL on a Web site requires special software, authentication certificates (usually issued by third parties, and requiring the payment of fees), and special configuration changes to Web servers. For this reason, it is used almost exclusively in cases where money transactions are taking place on a site, such as online shopping or banking. The average webmaster doesn't usually have a use for SSL.

Another protocol, TLS, is a newer form of SSL-like functions, but it is not as widely used as SSL for the Web (it sees more use in e-mail).

Note that SSL and TLS do not restrict the accessibility of Web pages; they can still be accessed by anyone. They simply protect the contents of the pages as they are passed to visiting clients, and they protect the information sent by the clients to the server, if any. For obvious reasons, the pages protected by SSL are often dynamically generated; that is, they contain confidential information specific to each visitor. A page that is static and available to everyone need not be protected by SSL, since anyone who wants to see it can do so, anyway, and nothing on the page is specific to each visitor.

#### 4.3.11 Protected Pages and Sites

By default, a Web site is completely visible to all visitors. Every page is freely accessible if a visitor requests it. It's possible to "hide" pages by not giving anyone the URL to the pages and not putting links to the hidden pages on any other known pages, but this is not a very reliable form of security, as the hidden pages may still be "discovered" by one way or another.

If certain pages or parts of a Web site must be protected against public disclosure, most Web server programs allow for password-based protection of all or part of a site. The Web server is simply told that certain pages, or groups of pages, cannot be accessed by visitors unless they first give a user name and password. The webmaster chooses the user names and passwords and issues them to visitors who are to be authorized to see private portions of the site. Different pages and areas on the site can be limited to different groups of visitors. This is a highly secure method of restricting access because it is controlled by the Web server itself, and if a visitor does not give the correct password and user name for a page, he won't be allowed to access it.

Note that this type of protection limits access to selected pages, but it does not protect the confidentiality of those pages as they are sent to the browser. To protect the confidentiality of information as it moves from server to client, or *vice versa*, over the Internet, SSL is required.

### 4.4 Browsers and Page Rendering

As we've seen, the actual Web page itself is just plain text with a few HTML tags thrown in. The nicely formatted version of the page you see in the browser window is created by the browser, based on the actual text in the page plus all the instructions it gets from the HTML tags (if any). When the browser *parses* (examines) the text of the page and acts on the instructions provided by the tags, it is said to be *rendering* the page for viewing. The way a browser renders a Web page is a function of the Web page itself and the tags it contains, the settings chosen for the browser (controlled by the user of the computer on which the browser is running), and on the design of the browser itself.

Good Web browser programs implement a set of standards set forth by the World Wide Web Consortium (W3C) for HTML rendering. Any browser that adheres to these standards will render a Web page in the same way, under the same conditions. Most of the most popular browser programs today conform closely to the W3C standards. Some very old browsers, however—such as the old Netscape 4.x browser and the oldest versions of Microsoft Internet

Explorer—seriously diverged from the standards and rendered many Web pages in unpredictable and very idiosyncratic ways. Fortunately, these older browsers are quite obsolete today and it is very rare for anyone to use them (the modern descendants of these browsers conform to standards correctly).

When the Web was first conceived, it was designed such that anyone could create a Web page with very little training. As a result, all of the HTML tags are essentially optional. If a Web page contains only plain text, with no tags at all, a browser will still display it, as plain text on the screen. If the page contains tags, the browser will attempt to interpret them in order to render the page as the author intended. Most importantly, though, modern browsers are designed to be extremely tolerant of improper use of HTML tags. If tags are missing or incorrectly coded, browsers will attempt to correctly guess how to render the page, and will do the best they can. This means that even very messy and incorrect HTML in a page can still produce something quite readable on the screen ... although of course it's always preferable to get the HTML right in the first place.

Another important characteristic of browsers is that they must render Web pages flexibly based on what the client computer can do. For example, the size of the screen on a computer varies from one computer to another, and a browser program must adapt the rendering of Web pages so that they come as close as possible to what the author requests no matter what the constraints of the client computer's screen. A typical example of this is the way that browsers position text. Text in a Web page is “reflowed” to fit the browser window in which it appears. The author of a Web page can control where new paragraphs begin, but the exact length of each line depends on the browser and the size of the browser window into which the page is being rendered. The final rendering is a compromise between what the author requests and what the client computer can do.

The combination of a browser's tolerance of errors and its adaptation to the client computer's local constraints mean that it is very easy to create Web pages that display nicely on the screen.

## 4.5 URLs

When a browser requests a Web page or file from a Web site, it identifies the page or file with a *Universal Resource Locator*, or URL. The URL identifies the computer that contains the page or file, the protocol used to obtain it (normally HTTP, for the Web), the exact location of the page or file, and a few other optional bits of information.

A typical Web-page URL looks like this:

`http://www.atkielski.com/main/Introduction.html:80`

The first part of the URL identifies the “type of resource”; for Web pages, the type is always `http`. The part of the URL between the first pair of slashes and the next slash is the fully-qualified domain name of the computer that holds the desired Web page (`www.atkielski.com` in this case). The rest of the URL up to the colon identifies the exact location of the page on the Web server. (The names of most Web pages end in `.htm` or `.html` by convention, but it's not mandatory.) The number after the colon is the port number to use in contacting the Web server; if this is not specified, the default port for Web pages is 80 (the well-known port number for the HTTP protocol).

Most browsers allow you to specify only part of a URL, and they will fill in the rest. For example, if you don't type `http://` at the start of the URL, the browser will assume this and will add it before actually accessing the Web page. If you don't specify a port number, the browser will assume port 80. If you specify a computer name but no location (or an incomplete location ending in a slash) for the desired Web page, the browser will ask the corresponding Web server for its “default” Web page; in this case, it's up to the Web server to decide what to send as the default page—in many cases, the default page for a Web server like `www.xyz.com` will be `www.xyz.com/index.html`, or `www.xyz.com/default.htm`, or something similar. The default page is often called the *home page* of the site.

Most browsers will also display the complete URL of a page after retrieving it, even if you only specified part of the URL. This allows you to see what the browser actually sent to the Web server when it requested the page. Thus, `www.xyz.com` may become `http://www.xyz.com/index.html` after the browser retrieves the page.

Because most Web pages contain images or other multimedia content, a browser will typically request more than one file from the Web server in order to display a page. The URL you enter in the browser and the URL displayed by the browser point to the actual Web page itself (the text file containing the HTML tags for the page), but the browser will also request any image files mentioned in `img` tags inside the page. This is why you often see a Web page partially rendered in the browser window, followed by the images on the page appearing one by one as the browser fetches them from the Web server.

## 4.6 Web Servers

Every Web site on the Internet must reside on a physical computer somewhere. The computer that holds the contents of a Web site is called a Web server, and is said to be *hosting* that site. A single Web server can host one site or many.

### 4.6.1 Web Server Hardware

In terms of physical hardware, the basic architecture of Web servers is generally the same as that of an ordinary

desktop computer. When differences exist, they usually reflect the requirement for Web servers to operate continuously and reliably over long periods. High-level Web servers are often very expensive because of this optimization for reliability and continuous heavy use, but it is also possible to run a Web server on an inexpensive PC—indeed, for small Web sites that have only a modest number of visitors, even an old, used PC can do the job.

The optimizations and modifications made in Web servers, as compared to desktop machines, include better, faster disk drives; better ventilation (more fans, larger fans); very high-speed network connections; power conditioning and battery back-up systems (often referred to as *uninterruptible power supplies*, or UPS); inexpensive video boards (servers never need fancy graphics, so even a cheap video board is sufficient); and so on. Instead of spending money on stereo speakers, large monitors, or expensive video and audio cards, server operators spend money on better back-up and ventilation systems, top-quality disk drives, redundant hardware, and the like.

It should be emphasized that nothing fundamentally prevents a Web server from serving as a desktop, or vice versa. The hardware is different in many cases only because of the heavy and repetitive loads placed on Web servers and the need to maintain 100% uptime in mission-critical roles. At the same time, since Web servers are often left unattended for weeks, there is no need for elaborate graphics capability, fancy audio hardware, or similar enhancements that might be routine for a desktop machine.

#### 4.6.2 Web Server Software

In terms of software, the salient feature of a Web server is that it runs a special *Web server program* (often called a *daemon*, following UNIX terminology) continuously in order to “listen” for Web requests from the Internet and respond to them. In most cases, the Web server also runs an operating system different from that of typical desktop computers.

Whereas most desktop computers run the Microsoft Windows operating system today, most Web servers run some version or clone of UNIX, an old but time-proven operating system specifically designed for servers. The most common operating systems for Web servers are thus Linux, a very popular clone of UNIX; FreeBSD, an open-source version of UNIX; Sun Solaris, a proprietary version of UNIX that runs on server hardware sold by Sun Microsystems; and Microsoft Windows, a special version of Windows optimized for servers, and the only one of this group that is unrelated to UNIX.

The most popular Web server program or *dæmon* today is Apache, an open-source server program designed to run primarily on UNIX systems, although it will also run under Windows. In a distant second place is Internet Information Server, a proprietary server program written by Microsoft that runs exclusively under Windows.

The largest Web sites run servers dedicated exclusively to hosting the sites; some even run multiple servers for a single site together with a mechanism for distributing the load equally among all servers. Smaller organizations may combine the Web server function with other server functions (e-mail, file transfers, chat or messaging systems, etc.) on the same physical hardware. It is even possible to run a Web server on a computer that is also being used as an ordinary desktop, although it's not a good idea to do so, because the needs of Web servers and desktop client computers are very different, and because Web servers must be carefully configured and operated in order to keep them secure (since they must be open to the Internet in order to do their jobs).

## 5 Building Your Web Site

Building a Web site is a matter of creating content, which in turn is mostly a matter of writing text and (if necessary) preparing pictures. When Tim Berners-Lee created the Web back in 1990, one objective of the project was to make it easy for non-specialists to create Web pages. This objective is still reflected in the Web today, and it is still quite easy to author and develop a Web page, even without special computer skills, provided that one starts out with something modest and works up from there.

### 5.1 Design Considerations

What to put on your Web site is of course your decision exclusively, but there are some general design principles that may help you in creating a Web site that people will want to visit, and we cover some of the more important among them here.

#### 5.1.1 Visitor Paths to Your Site

There are only two ways in which a visitor can reach your Web site: (1) she can type the URL of your site (or a page on your site) directly into her browser, or (2) she can click on a link on some other page that leads her to your site.

The first way of reaching your site implies that a visitor has somehow obtained the URL of the site or of a page on

the site by some means exterior to the Web—perhaps by reading your business card or letterhead, or perhaps simply because you told her the URL of your site. Most sites receive relatively few visitors by this path.

The second way of reaching your site is much more common. And, in most cases, the page providing the link to your site will be a page produced by a *Web search engine*. Search engines are simply Web sites, such as the archetypal Google, that allow visitors to scan an index of other Web sites in search of sites that correspond to specific keywords or other criteria. In some cases, the link to your site will be on some other site on the Web (other than a search engine site). If you advertise your site on the Web, visitors may arrive on your site by clicking on advertisements for the site appearing on other sites.

Nobody stumbles onto a Web site by typing a random URL. This is important to understand, because it means that anyone reaching your site already wants to see the contents of your site. A consequence of this is that your site should not be self-promoting. In other words, you don't need advertising or promotional copy on your site to tell visitors what a great site it is to visit—because they've already decided to visit the site by the time they arrive.

### 5.1.2 Navigation

Speed and convenience are everything for Web surfers. One way to help ensure that your visitors linger a while on your site is to make navigation of the site fast and simple.

The standard way to navigate around a site is by clicking on links within the site, and this is the simplest and fastest way to navigate. A simple table of contents or site map with links to all useful pages isn't a bad idea, particularly for a new site. A good way to test the navigation on a site is to visit the site with a text-only browser such as Lynx. If you can navigate a site successfully and easy with Lynx, the navigation is probably well designed.

### 5.1.3. Graphics and Color

There are still some books and other resources on Web design that make outdated recommendations with respect to graphics and color on Web sites. Today, the restrictions they suggest are no longer pertinent.

For example, in the old days, it was recommended that GIF be the preferred type of image file used on Web sites. GIF is a file format originated by CompuServe (now part of AOL) for images using on their computer serve. Early browsers handled only GIF files correctly. However, GIF is outmoded today, and the more modern JPEG format is preferred for most images, particularly photographs. GIF requires more space, allows only 256 colors (instead of the millions of colors permitted by JPEG), and cannot accurately represent smooth gradations of color. It is still suitable for very small images containing only a small, fixed number of colors, but beyond that it is obsolete.

Another early recommendation for Web design was to use only “Web-safe” colors in the design. This recommendation was based on a combination of limitations imposed by the early Netscape browser, early versions of Microsoft Windows, and early PCs, which together made it impossible to display more than 256 distinct colors on the screen of a computer at once (including all intermediate shading, etc.). The Web-safe colors ensured that no bizarre color-shifting occurred on the screen. Today, however, all PC hardware and all versions of Microsoft Windows handle up to four billion colors without any trouble at all, and the old Netscape browser is history as well, so there's no longer any reason to restrict the number of colors used on a Web site, at least from a technical standpoint.

Nevertheless, there are aesthetic considerations that may limit and amount of colors used on a site. Web sites with a vast number of intense, clashing colors are the hallmark of the amateur. As in any other area of design, a limited palette of tasteful colors is usually preferable to a Technicolor rainbow that causes eyestrain among visitors. In general, the same rules that make for pleasing printed materials also apply to Web sites. For example, except in very special cases, it is customary and advisable to stick with dark text on a light background for Web sites, as this is less hard on the eyes, particularly when a Web surfer is moving from other Web sites (most of which are black-on-white) to your Web site.

When it comes to images, the situation is a lot more flexible than it was in the days of Web-safe colors. You can generally put images on a Web page in GIF, JPEG, or even PNG formats, although the first two are safest (there are still a few browsers around that cannot display PNG images correctly). Other image formats are not supported by enough browsers to be safe for Web sites—too many visitors might not see the images correctly if they aren't GIFs or JPEGs. GIF is handy for very small images with a limited number of individual colors and no smooth gradations, as we've said above; JPEG is suitable for everything else.

GIF files have a feature that allows you to animate the images they contain. Animation, like rainbow colors, is one of the hallmarks of the amateur on the Web, but occasionally an extremely limited use of animation might be appropriate. Note that animation can be achieved by client-side scripting as well as the use of GIF files.

Another consideration with graphics and images is download time. Text doesn't take up much space and downloads very quickly on a Web page, but images can be extremely time-consuming to download if they are too large, too numerous, or incorrectly prepared. Be very careful about the images you put on your Web site, or it may take so much time to download that visitors will leave without waiting for the download to finish.

### 5.1.4 Dynamic Content

Dynamic content, as we have explained in Section 4, is content that is generated at the time a Web page is visited, either on the Web server or on the client computer. Including dynamic content is a fairly ambitious undertaking for a beginning webmaster, but it may have some utility in moderation for certain types of sites and in certain situations (a classic example being the visitor counter that appears on many personal Web sites).

If the dynamic content is generated entirely on the server, as through server-side scripting or CGI programming, then it appears identical to static page content from the viewpoint of the client computer, and has little influence on design considerations. If the dynamic content is created on the client side of the interaction, then a number of factors need to be taken into consideration in the design of the site.

One major consideration is whether or not client-side dynamic content can be supported by the browser. For example, over 99% of all browsers today (in terms of market share) support client-side scripting in Javascript, and so this type of dynamic content can usually be safely employed on a Web site. In contrast, dynamic content that requires specific ActiveX controls or plug-ins is much less likely to be supported by visitors' computers. For example, less than half of Web surfers have support for Real Audio on their computers, and so making Real Audio an integral part of a Web site risks driving away more than half of the visitors to the site. If you must include this type of dynamic content on a site, it's best to design the site so that it will work reasonably well even if a visitor does not have the means to exploit the dynamic content; in other words, if you must have a Flash animation on your site, try to design alternate pages on the site for visitors who do not have Flash installed or available.

Some Web surfers have the capability to handle dynamic content, but prefer not to, for security reasons. For instance, some surfers turn off all ActiveX or plug-in features in order to prevent viruses or other malware from being installed on their computers (dynamic content is an extremely common vector for transmission of such malware). These surfers can be assimilated with those who don't have the capability in the first place. It's a good idea to provide for them, particularly if the dynamic content in question is not universally supported; *i.e.*, while it is usually safe to assume that everyone has Javascript enabled and available, the same is not true for Flash or Real Audio content—so it might not be vital to develop a site that will work even for browsers that don't support Javascript, but it would be important to develop a site that would work even in the absence of Flash support.

### 5.1.5 Accessibility

Accessibility is the ability of a Web site to accommodate visitors with various disabilities. In the world of the Web, the disabilities that most often raise design issues are vision impairment and hearing impairment, especially the former.

People with severe vision impairments may use special, text-only browsers such as Lynx to access a Web site, or they may use screen readers in conjunction with normal browsers. Some special-purpose browsers for the vision-impaired combine both functions in one program. Because of this, a site must be readable and navigable in a useful way even for browsers that cannot display images. One way to verify this is by visiting the site with a text-only browser. Another way is to have the site tested for accessibility by services such as the Center for Applied Special Technology (CAST), <http://www.cast.org>.

The key points to remember when designing for accessibility with respect to vision-impaired users are that all images must have captions, and nothing on the site should *require* the ability to display or see an image (unless the image is the essence of the content, such as a photo album—even here, captions can be used to make the album useful even for visitors who cannot see).

For hearing impairments, the accommodations are much simpler. As long as the site does not depend on audio content for navigation or other purposes, it should not be much of a problem for the hearing-impaired. Relatively few sites have large volumes of audio content, much less audio content that must be heard in order to navigate the site, and so there are few problems with this type of accessibility.

Most other disabilities are transparent to webmasters, and don't require any special accommodation in Web design.

### 5.1.6 Download Time

Download time is the time required to transfer the contents of a Web page from the Web server to a client computer. It's a function of the size of the contents of the page and the speed of the connection between the client computer and the Web server.

If the pages of a Web site contain only text, they are usually not very large in size, and download times should be very short, even for visitors with slow connections. The real problem arises when a Web page contains a lot of graphic elements or images, or active content such as Flash animations or downloadable ActiveX controls. For this reason, the use of images and graphics should be weighed carefully against the increase in download time that they represent.

If sharp detail in an image is not essential, it can be delivered in the form of a JPEG file with high compression, which can save a great deal of download time. GIF files occupy more space (particularly for images with few large

areas of constant color) and have only a single level of compression, and thus should be avoided for large images that contain more than two or three colors in simple arrangements.

Some Web server programs allow for automatic compression of Web page content before it is sent to the client computer. This can save considerable download time at the expense of a somewhat greater load on the Web server. It is also necessary that the client computer's browser recognize and accept compressed content, but most modern browsers do.

### 5.1.7 Handwritten HTML vs. Authoring Tools

It is perfectly possible to create Web pages using a simple text editor, and many sites are still created this way today, including some very large sites. This is so because the original design of the Web was intended to allow non-specialists to create usable Web pages without any special tools. Even today, some surprisingly large sites are still created by webmasters who type the text of their Web pages (HTML tags and all) manually using ordinary text editing programs.

For those who don't wish to type Web pages and HTML tags by hand, there are Web authoring tools, such as Microsoft's FrontPage and Adobe's GoLive. These tools are very expensive, elaborate products that allow for WYSIWYG (*what you see is what you get*, pronounced *whizzy-wig*), meaning that one can simply "draw" pages on the screen of a computer using the authoring tool and the tool then generates the actual Web page text and HTML tags.

HTML written by hand is usually fairly neat and clean, although this depends on the human author to a large extent. HTML generated by authoring tools is often somewhat bloated and messy, since the authoring tools assume that a human being will not look at the HTML directly and include a great many tags, comments, and other material that are not strictly necessary for the Web page in question, but are too difficult to selectively delete based on page content. In some cases, the generated Web pages are so complex that it is extremely difficult to modify them after hand once they have been generated. In addition, most authoring tools work best with Web pages they have generated themselves, although they can read and modify pages generated by hand (or by other authoring tools).

Beginning webmasters can profit from learning the use of HTML tags and making use of them to create Web pages by hand, since a good grounding in the types of tags available and the ways in which they can be used makes it easier to design Web pages that look as the webmaster intends them to look and are not unnecessarily large. Experienced webmasters sometimes continue to prepare critical Web pages by hand, in order to minimize their size, control the details of their content more carefully, or provide for situations that are not handled by authoring tools.

## 5.2 Hardware Requirements

The computer hardware you need to create your own Web site is minimal and not really any different from the computer you might already be using at home or office. Indeed, as long as you have a computer with an Internet connection, you probably don't need any additional hardware in order to start building a Web site.

### 5.2.1 Client Computer

A client computer is simply the computer on which Web pages are created, as opposed to the computer that hosts a Web site and answers requests for Web pages from that site.

Just about any computer can be used to create Web content. All that's needed is a text editing program and a program that can transfer the finished Web pages from the client computer to the Web server. Standard desktop or laptop PCs or Macs are more than adequate for this task.

### 5.2.2 Internet Connection

The client computer used to create Web pages needs an Internet connection so that it can upload finished pages to the Web server that hosts the corresponding Web site.

Since the Internet connection of the client computer only serves to upload Web pages to the actual Web server on the Internet that hosts the Web site, just about any type of Internet connection will do.

#### 5.2.2.1 Dial-Up Connections

*Dial-up* connections to the Internet are temporary connections that are established via an ordinary telephone line, using a device called a *modem* (short for *modulator-demodulator*). The modem translates data from the client computer into sound, which is then sent over the telephone line as any other sound would be. Another modem at the other end, installed in a computer belonging to the Internet Service Provider, changes the sound back into computer data, which is then routed over the Internet. The inverse process is carried out to send data from the Internet back to the client computer.

Dial-up connections are slow but inexpensive, and they can be used wherever telephones are available. The dial-up connection is not permanent: it is established by making a telephone call with the modem to the ISP when Internet access is required, and then the call is terminated when access is no longer required.

### 5.2.2.2 ADSL Connections

*ADSL* connections (*ADSL* is short for *asynchronous digital subscriber line*) are high-speed, permanent connections to the Internet, part of the category of Internet connections commonly referred to as *broadband*. *ADSL* is an increasingly popular way for homes and small offices to connect themselves permanently to the Internet, but it is only available in built-up areas, for technical reasons (*ADSL* connections don't work over the long distances encountered in rural areas).

An *ADSL* connection uses the same telephone line that is used by a dial-up connection, but it does not convert data to sound. Instead, it uses inherent, spare capacity in the actual copper wires connecting the local telephone to its nearest telephone exchange to transmit data in a special form at extremely high speeds—up to 30 million bits per second (one thousand times faster than a dial-up modem). *ADSL* connections are permanent, 24-hour-a-day connections, and they do not interfere with normal telephone calls (a dial-up connection occupies the telephone line for the duration of the connection, making voice calls impossible).

### 5.2.2.3 Cable and Other Broadband Connections

It is also possible to permanently connect a home or office computer to the Internet using spare capacity in the same cable used for cable television. *Cable* connections are more sensitive to traffic, sometimes running very fast and sometimes running very slow, and they are mostly limited to home installations, since few offices have existing cable TV connections. They do not require a telephone line, unlike *ADSL*, but a cable TV connection is mandatory.

*ADSL* and cable are the primary broadband connections used for homes and small offices. Other technologies are in the works but are not widespread today.

### 5.2.2.4 Internal LANs

Large companies often have internal local area networks, or *LANs*, which are permanently connected to the Internet. If you are building a Web site from within such a company, you may be able to use the corporate *LAN* to connect to the site in order to upload your Web pages. Consult with your internal computer support staff for more information.

### 5.2.2.5 Wi-Fi Hotspots

It's possible, albeit awkward, to use wireless *Wi-Fi hotspots* to connect to the Internet for uploading Web pages to a Web site. Since all *Wi-Fi* connections are temporary, however, this isn't a very convenient way to work, unless the *Wi-Fi* hotspot is installed directly in the home or office where the webmaster normally works. *Wi-Fi* connections are connected in turn to internal *LANs* or broadband connections for their actual access to the real Internet.

## 5.2.3 Miscellaneous Hardware

No special hardware is required to create and maintain a Web site. If images need to be manipulated for use on the site, a graphics tablet is handy (particularly if photo retouching is required), although it is not essential. None of the operations involved in creating a Web site requires a particularly fast or performant computer.

## 5.3 Software Requirements

Building a Web site requires only a few simple software tools, many of which are already found on the average PC or Mac. It's possible to spend far more money on special Web-authoring products, but such products offer mainly convenience—they don't make possible anything that one cannot already do with simpler tools.

### 5.3.1 Text Editors

Since Web pages are essentially text files (with a few HTML tags thrown in), it follows that a simple text editor should be sufficient to create them ... and in fact this is precisely the case.

On a Windows computer, just about any text editor will do, including the built-in Windows Notepad text editor. Word-processing programs like Microsoft Word can be used as well, but it's important to save files in plain text format if one is using a word-processing program, otherwise they won't work correctly as Web pages. For word-processing programs (including Word), there are add-ons or plug-ins available that can save documents in Web-page format (text with HTML tags), but this option often produces only mediocre Web pages. There are also freeware, shareware, and commercial text-editing programs specifically designed for tasks like building Web pages, and these can be a very good value; a typical example is IDM's UltraEdit-32. When it comes to text editors, there's no shortage of choices, with something for every taste and wallet. Most text editors that are not free are in the €40 range.

One issue that you should keep in mind is the coding of the text file. For letters, digits, and other ordinary characters in a text file, the computer and editor that you use are of little importance; but if you plan to use special characters in a Web page, such as copyright symbols, accented characters, and the like, you need to be sure that you use the same computer environment to create the text file that will be used by most of your visitors to view the result-

ing Web page. This is necessary because different computers and different environments code the special characters differently. If you build your page on a Mac and include special characters such as accented letters (for pages in non-English languages, for instance), the special characters might not look right when the page is visited by people using Windows computers.

If you really must build “cross-platform” pages in this way, you should use HTML *escape sequences* to encode special characters. An escape sequence is a string of characters that unambiguously identifies a special character to a browser. For example, when a browser sees the escape sequence `&copy;` within the text of a Web page, it replaces the sequence with a copyright symbol ‘©’ when it displays the page on the screen ... no matter which type of computer the user of the computer has. If you are creating your pages on a Windows computer with a Windows text editor, you may be able to get by without using escape sequences, since 95% of your site visitors will also be using Windows to view the site; but if you are using any other platform to create your pages (Linux, Mac, etc.), it’s safer to use escape sequences for special characters (including accents on letters).

### 5.3.2 FTP Clients

You need a way to transfer your finished Web pages from your computer to the Web server that hosts your Web site. The standard way to do this is with an Internet protocol call *file transfer protocol*, or FTP. Therefore you need a program that “speaks” FTP, called an *FTP client*.

Many computer systems, including Windows systems, offer a built-in FTP client, but it is usually so awkward to use that it’s not very practical for building a Web site. Typically you’ll want to acquire an external FTP client for transferring your Web pages, and, as with text editors, there are hundreds to choose from, such as CuteFTP, WS\_FTP, AbsoluteFTP, and many others. Some are oriented towards Web building, others are not. Prices are in the same range as those of text editors, at around €40.

An FTP client is very simple. It allows you to move a file from your computer to your Web server, and vice versa. Some FTP clients integrate directly into Windows Explorer, allowing you to drag and drop files to and from your Web server as if it were a disk drive (of course, this requires an Internet connection). More traditional FTP clients usually display two windows, one on the Web server, and one on your computer, and allow you to drag and drop files between them.

### 5.3.3 Browsers

A *browser* is a program that runs on a client computer and “speaks” the HTTP protocol over the Internet in order to obtain information from Internet Web sites. You need a browser program on the computer you use to build your Web site so that you can actually visit the site and check to see that Web pages are being rendered correctly.

Many computers are supplied with a built-in browser of some kind, but most computers also allow the user to install and use the browser of her choice. Popular browsers for desktop computers at the time of this writing include Microsoft Internet Explorer (the leader by far), Firefox, Safari, Opera, Mozilla, and Konqueror.

A browser initiates a connection with a Web server on the Internet when the user of a computer starts the browser program and enters the URL of a Web site or Web page. The browser finds the Web server containing the desired site or page and connects to it directly, then requests the page or pages desired using the HTTP Internet protocol. When the pages are received from the Web server, the browser scans the contents of the pages looking for HTML tags, and interprets any tags that are present. The result of this interpretation, the *rendering* of the page, is displayed in the browser window on the screen of the client computer.

The design of a browser has a tremendous influence on the way that Web pages will appear on a computer. Five different browsers might display the very same page in five different ways, although most of the best and most popular browsers conform to standards for rendering pages established by the World Wide Web Consortium. When highly conformant browsers are used to visit a site, they will render the pages on the site almost identically. Less conformant browsers may render pages in surprising ways at times.

To test your Web pages, you should use a browser that is the same or equivalent to the browser(s) used by the majority of visitors to your site. At the time of this writing, over 90% of Web surfers use Microsoft Internet Explorer to visit Web sites, so you should either test your Web site using this browser, or test the site using some other, equally standards-conformant browser (such as Firefox). In theory, any Web site that conforms to W3C standards in its use of HTML will display in a predictable way on any browser that conforms to these standards as well. In practice, finding a browser that actually did conform to the standards was quite difficult in the past (the older Netscape Navigator was famous for the number of errors it contained with respect to the standards), but today most major browsers are highly conformant.

### 5.3.4 Image Editors

Although there’s no requirement that a Web site include images or other graphic elements, the vast majority of Web sites do. If your site will contain photos, diagrams, or any type of visual content other than text, you’ll probably need some sort of image-editing program to prepare the content.

The Rolls-Royce of image editors—and the professional standard for image editing—is Adobe Photoshop, and it doesn't really have any direct competition. However, it's very expensive, at several hundred euro, and it provides features that are not really necessary if your only intention is to prepare a Web site (for example, it contains a lot of features that are specific to the world of commercial printing). If your tastes or budget run more towards cookies than caviar, there are programs such as Paint Shop Pro that cost considerably less and still provide all the functionality you need to prepare images for the Web. At the extreme bottom end of the scale are modest, built-in programs such as Microsoft Paint (provided with Windows), but they are so primitive that they usually are inadequate for webmasters.

Image editors let you create images for your site from scratch, but more importantly, they allow you to modify images so that they are suitable for Web use. In particular, it's often necessary to modify the size of an existing image, or compress it, or change its file format before uploading it to a Web site. If you have a photo from your digital camera that you'd like to put on your Web site, for instance, you typically cannot upload it to the site as-is, because it's too big; you must modify the size of the image, compress it a bit, and save it in an appropriate format (JPEG or GIF) before moving it to your Web site. This requires an image editor.

Some image editors are bundled with other products, and they may be suitable for Web use. Many digital cameras, scanners, etc., are provided with software that includes some sort of image editor. Very expensive equipment (in the multi-thousand-euro range, such as professional digital SLR cameras) may even come with a bundled copy of Adobe Photoshop.

### 5.3.5 Web-Authoring Tools

A vast number of sophisticated Web authoring are available for creating Web sites today. Some of the most popular include Microsoft FrontPage, Netscape Composer, Adobe GoLive CS, and Macromedia Dreamweaver MX. Most of these are WYSIWYG products. Some of them require special software on Web servers.

A Web-authoring tool is usually an all-in-one product that combines a sophisticated, Web-oriented visual page editor with functions for organizing and uploading finished pages to a Web server, as well as support for testing and viewing pages before and after they are moved to the server. A typical tool of this kind takes the place of all or most of the other tools we've mentioned above (text editor, FTP program, browser, image editor, etc.).

Web-authoring tools have the advantages of comfort and advanced features: everything is designed to work together, so only one product has to be installed on the client computer in order to create Web sites. However, they are also very expensive, often very bloated (requiring lots of disk space and memory on the computer), and sometimes they lock you into one specific product, by making it difficult or impossible to edit a Web site with any other Web-authoring tool without significant changes to the site. As previously mentioned, they may also require special software on the Web server. Some tools may not allow you to actually see the Web-page text and HTML tags directly, or they may generate text with such complicated tags that it can be extremely difficult to understand or edit the content directly, without going through the Web-authoring tool's normal interface.

Generally speaking, you ultimately must choose between building your Web site directly using the simple tools previously described (text editor, FTP program, and so on), or purchasing a Web-authoring tool to provide an all-in-one solution for creating the site. Web-authoring tools are good for developing moderately sophisticated sites quickly, but they do prevent a webmaster from being exposed directly to Web-page HTML, which may not be a good thing. And for mission-critical sites, the simple tools are often preferred by seasoned webmasters over Web-authoring tools, for reasons of simplicity, reliability, interoperability, and flexibility.

### 5.3.6 Dynamic and Active Content Considerations

If you plan to include dynamic content in your site (perhaps a bit ambitious for a newbie webmaster, but it's up to you), you may need to invest in development tools, depending on the programming or scripting languages that you plan to use. Scripting languages such as PHP or Javascript require only a text editor in most cases, but more complex languages, especially true programming languages such as Perl or C++, require *compilers* and/or *integrated development environments (IDEs)* that permit you to code, develop, test, and organize your scripts or CGI programs easily. IDEs and compilers cost money, and they often are not cheap. If you are using an open-source operating system such as FreeBSD or Linux, however, you may be able to develop what you require with freeware development tools.

### 5.3.7 Operating-System Considerations

The operating system of a computer is the software that gives it its basic "personality" or "look and feel." The most popular operating systems for desktop computers are those of the Microsoft Windows family, but there are many others, including the Apple Mac OS family, Linux (which comes in several hundred different flavors), IBM's aging OS/2, and so on.

The operating system of the computer you use to create your Web site is not very important; you can design and develop a Web site using just about any operating system on any computer. As long as the operating system sup-

ports the tools you need (the aforementioned text editors, FTP programs, and so on), you can use it to create a Web site ... and most operating systems easily support all of these tools, since they are quite generic and were fixtures on most computer systems long before the Web came along.

If you are planning to use a Web-authoring tool to create your Web site, the tool you choose may constrain your choice of operating system, or *vice versa*. For example, Microsoft FrontPage isn't available for Linux, so if you have your heart set on using FrontPage, you can't run Linux as your operating system, and if you are bent on running Linux as your operating system, you'll have to choose some other Web-authoring tool instead of FrontPage.

The operating system of the computer that hosts your Web site—the Web server—must be selected more carefully, but we'll cover that later in this paper.

## 6 Publishing Your Web Site

The creation of your Web site is something you do on your own computer, at your home or office. The publication of your Web site is the act of making it visible and available to the rest of the world on the Internet, and it is distinct from creation of the site.

A Web site is normally published by placing the contents of the site on a Web server, which is a computer that runs 24 hours a day and is permanently connected to the Internet. The Web server is said to *host* the site. The Web server actually receives visitors to the site, and responds to their requests to view individual pages on the site. While it is theoretically possible to create *and* publish a Web site on the same computer, it's not a very good idea, for various reasons, and in practice the two activities always involve separate computers. For this reason, we discuss creation and publication of Web sites in two separate sections of this paper.

Sections 6.2 and 6.3 below describe the hardware and software requirements of a Web server. They are essential information if you plan to host your own site with your own computer, but if you plan to have your site hosted by someone else, you can skip these sections.

For most new webmasters publishing their first site, external hosting is the best choice, which is why we cover it first.

### 6.1 External Hosting Options

In most cases, it's more practical to have someone else host your Web site than it is to try to host it yourself. *External hosting* means having someone else run a Web server on your behalf, which hosts your Web site. External hosting is usually very affordable and it requires far less technical expertise and time on your part than would be required if you chose to run your own Web server. Most webmasters choose external hosting, unless they are running very large sites or are very geeky.

External hosting comes in several flavors. In general, you get what you pay for. The more you pay, the more control you have over your Web site, and the more options, features, and flexibility you have in its design and operation.

#### 6.1.1 ISP Hosting

Many Internet Service Providers offer some sort of Web hosting services as part of their standard Internet subscriptions. This is a very inexpensive way of publishing a Web site, but it is also very restrictive.

When an ISP provides Web hosting, it often amounts to setting aside a spot on the ISP's main Web server (or a server dedicated to subscriber Web sites) with a fixed URL that incorporates the subscriber's name, *e.g.*, a URL of `www.bigisp.com/subscribername`. This works, but it's not very individualized; it might well do for a personal Web site, but it looks tacky for the Web site of a business or sole proprietor.

In addition to setting aside only a limited amount of space for each subscriber Web site, ISPs often place other significant restrictions on the sites they host: no dynamic content (no server-side scripts), no advertising or business use, limited bandwidth (that is, a limit on the amount of traffic the site is allowed to receive before the ISP starts charging extra for it), restrictions on the type of content that can be on the site (no nudity, no politics, no religion, or whatever), and so on. Some ISPs also require that subscribers update and maintain their sites using only proprietary software tools provided by the ISP or by a third party—they may require that subscribers use a special home-cooked program to upload Web pages (instead of a standard FTP program), or they may support publishing of pages only through Microsoft FrontPage software. In the case of Web-authoring tools that require special software on the Web server, many ISPs may refuse to install the necessary software on the server side.

Since ISPs provide Web hosting as a fringe benefit of their basic service (which is providing access to the Internet), they tend not to be as diligent about maintaining their Web servers as a dedicated Web-hosting company might be. The Web servers of an ISP are more likely to be overloaded or unavailable than those of a company that makes its

money by selling Web hosting. This may be an important consideration if availability and performance are crucial to your site.

Some ISPs also offer Web hosting as an extra service, for an extra fee. In this case, their product offer is similar to that of dedicated external Web-hosting companies, which we cover in further detail below.

### 6.1.2 Free Hosting

Some companies offer “free” Web hosting services. A classic example is Yahoo’s GeoCities (available at [www.geocities.com](http://www.geocities.com)). These companies specialize in Web hosting, and they provide it for free to subscribers, but they expect certain rights to the content of the sites that subscribers create in exchange for the free hosting, and they also reserve the right to advertise on subscriber sites.

As with ISP hosting, free hosting usually offers very little flexibility or control over a site. The URL is generally of the same general form as that used for ISP hosting, and services such as server-side scripting, CGI, support for Web-authoring tools, etc., aren’t available.

There are a few specialized types of Web sites that might be well suited to free hosting. For example, if all you wish to put on your Web site is a *blog* (an online journal or diary), there are many free hosting sites that provide all you need to create a blog site, and you don’t need any special software, or even a computer (you can do all the maintenance of the site from an Internet café, if you wish). Some other free hosting sites allow you to create online photo albums as well. If your only reason for creating a Web site is to publish a blog or photos or some other type of extremely circumscribed content, a free hosting site may be a very cost-effective solution.

Many free-hosting sites require that you use special online or downloadable tools to create and maintain your Web site, *i.e.*, you cannot use standard tools (such as those we’ve previously described) to do these things.

### 6.1.3 Web-Hosting Companies

Web-hosting companies provide and maintain Web servers on which you can build Web sites, for a fee. You have complete control over the content of your Web site, but you need not worry about maintaining a physical computer or Internet connection for it. (Remember that a Web site must be hosted on a Web server computer that runs 24 hours a day and is continuously connected via high-speed link to the Internet, with a static IP address.)

Web-hosting companies typically offer a wide variety of hosting plans, with one for every wallet. Often hosting fees start as low as a few euro per month, and can go as high as hundreds or thousands of euro per month.

We describe some of the most common hosting plans below. They are listed roughly in order of increasing price, capacity, and functionality.

#### 6.1.3.1 Shared Virtual Hosts

In *shared virtual hosting*, the Web-hosting company sets one or more common Web sites to be used by several of its customers. When you subscribe to shared virtual hosting, the Web-hosting company sets aside a portion of one of these common sites for your exclusive use. You don’t have your own domain name, but you may be given your own subdomain name. In other words, you can’t have [www.mydomain.com](http://www.mydomain.com) as your URL for your Web site, but you may be able to obtain [mydomain.webhostingcompany.com](http://mydomain.webhostingcompany.com) as the URL of your Web site.

Shared virtual hosting is usually the least expensive type of hosting plan, at a few euro per month. You are typically limited to static content on your site (that is, you cannot create server-side scripts or CGI), and you have little or no control over site configuration parameters such as password protection, SSL, and so on. There may be limits on the total size of your site and/or supplemental fees if you exceed a certain size, and there may also be limits or fees for *bandwidth* (the amount of data you send and receive from your site). You can typically access your site with standard tools, and very often there are provisions for specific web-authoring tools as well (such as Microsoft FrontPage). Shared virtual hosting normally does not support database-backed or e-commerce sites, except at a few Web-hosting companies that are specifically specialized in this type of site.

In technical terms, a shared virtual host means a single Web server that hosts multiple sites, each of which is shared by multiple webmasters.

Shared virtual hosting is often an excellent choice for new webmasters constructing their first Web site.

#### 6.1.3.2 Dedicated Virtual Host

*Dedicated virtual hosting* is similar to shared virtual hosting, except that an entire virtual host is dedicated to your site. This means notably that you can host your own domain, such as [www.mydomain.com](http://www.mydomain.com). It also gives you more control over site configuration and options such as SSL, password protection, and database support, although the Web-hosting company may require that you go through them in order to install or modify these features (and sometimes you may be asked to pay an additional fee for each such intervention). The one thing you may still not be allowed to have is server-side scripting and CGI, mainly because these features raise security issues that cannot be safely resolved in virtual host environments (that is, there is no way for the Web-hosting company to keep their

machines secure if they allow customers to use server-side scripting or CGI on virtual hosts).

### 6.1.3.3 Virtual Server

A *virtual server* is a substantial step up from mere virtual hosts. Using special software, a Web-hosting company divides a single large physical computer into several virtual servers, each of which behaves like a completely separate computer and has no knowledge of the other virtual servers on the same physical machine. Individual customers are then each given their own virtual servers, which they can manage as they see fit.

Virtual servers give you full control not only over your Web site, but also over the Web server, such that you can often set up your own e-mail handling for your domain (in simpler hosting plans, the Web-hosting company sets up e-mail parameters for you). You also have complete control over SSL, password protection, Web server configuration, and many other features. You can use CGI and server-side scripts freely, since your virtual server is completely isolated from all other virtual servers.

Managing virtual servers requires more technical knowledge than working with virtual hosts, although most Web-hosting companies will set things up for you initially, and you need not modify anything thereafter.

### 6.1.3.4 Dedicated Server

A dedicated server is a physical computer dedicated to your Web site. The Web-hosting company provides a facility for the computer, the computer itself, the Internet connections, and regular backups of the server, but you handle all the rest. It's like running your own computer, except that you don't have to worry about fire and power protection, backups, telecommunications charges, and other details associated with actually running a physical computer.

With a dedicated server, you typically have administrator access to the machine, meaning that you have all the system administrator privileges and can run and configure the machine however you wish. This arrangement gives you maximum flexibility for your site, with no particular restrictions on domain names, SSL, password protection, CGI, and server-side scripting. The Web-hosting company doesn't need to worry about security, since the machine is entirely dedicated to you, and security becomes your problem, not theirs, for the most part.

This is the most expensive type of Web-hosting, and it also requires the most technical expertise. Often you must pay a very high start-up fee (in some cases, the Web-hosting company may actually be buying hardware specifically for your site), and you may have a minimum length for your subscription, such as one or two years. You need to know exactly what you want, as in many cases the Web-hosting company will custom-configure a computer for you based on a variety of specifications that you provide.

## 6.2 Hardware Requirements

If someone else doesn't provide a computer to act as the Web server for your site, then you must do this yourself. Self-hosting simply means running your own Web server in your home or office.

This section covers the hardware requirements for your own Web server; if you plan to have your Web site hosted by someone else, you can skip this section.

### 6.2.1 Server Computer

A Web server computer is a computer that hosts a Web site (or several Web sites). It waits for client computers elsewhere on the Internet to contact it with requests for pages from the site(s) it hosts. Because the Internet is a worldwide network, a Web server must run 24 hours a day and seven days a week, and it must have a continuous, high-speed connection to the Internet.

Because Web servers are dedicated to a purpose very different from that of the average desktop client computer, it is extremely difficult (not to mention extraordinarily unwise) to try to combine the functions of both server and client computers in a single machine, even though this is technically possible. Without exception, for any production Web site, it is always preferable to keep the Web server computer(s) and the client computer(s) completely separate.

Although Web servers typically have fundamentally the same hardware architecture as ordinary desktop computers, server hardware is chosen with different goals. For example, servers often have more rugged components selected for reliability and performance, such as high-quality, high-speed disk drives, large and top-quality fans, and so on.

The degree to which this emphasis on reliability and performance is carried depends on how critical the Web site being supported by the server happens to be. For large, mission-critical Web sites with very heavy traffic (such as those of, say, an Amazon.com or a stock brokerage), the measures taken in order to ensure that the Web server does not fail can be extreme, with fully redundant hardware configurations, generator and battery backups for power, special disk arrays that can continue to operate even after multiple failures, and so on. However, for small sites with light traffic, just about any computer will do—even an inexpensive, second-hand desktop computer can be used as a Web server if the site is small. In fact, Web servers for modest Web sites can be even tinier than a typical desktop computer, since the demands on the hardware for a small Web server are considerably less than those placed on a

typical desktop computer.

This means, in turn, that if you are just starting to publish your own Web site, and you don't expect much traffic, you can use an old, second-hand computer as your Web server without too much trouble. A machine too outdated to provide acceptable performance for a modern desktop client computer may still be more than adequate as a Web server.

### 6.2.2 Internet Connection

Unlike a client computer, which can easily get by with a temporary and relatively slow Internet connection, a Web server requires a continuous, high-speed connection to the Internet. If you are running your own Web server, this means a broadband connection, such as ADSL or cable. You also need a static IP address, which may not be offered by all ISPs; ISPs that do offer a static IP address sometimes charge an extra monthly fee for it. ADSL broadband providers are more likely to offer static IP addresses than cable broadband providers.

If your site generates a great deal of traffic, or a large volume of data transfer (such as a photo album site might), you may grow out of standard broadband connections. In this case, you may be obligated to pay for much faster Internet connections that require special equipment, and the cost of telecommunications for your site can skyrocket. In fact, very often the limiting cost factor for webmasters who self-host relatively small sites is the cost of the Internet connection, and not the cost of the Web server itself. As long as the traffic generated by your site can fit in the capacity of a standard home or office broadband connection, you can self-host at very low cost. If the traffic exceeds that threshold, it might be cheaper to pay for external hosting.

## 6.3 Software requirements

The software required for a Web server is substantially different from that required for a client computer, right down to the operating system, which is one of the reasons why it is not a good idea to try to use the same physical computer as both a client and a server.

Here again, if you are interested only in having someone else host your Web site, you can skip this section.

### 6.3.1 Operating Systems

An *operating system* is the basic software that a computer runs “beneath” specific application programs. The operating system handles all the “housekeeping” tasks of a computer, provides a stable platform upon which application programs (such as text-processing programs, games, business software, browsers, or whatever) can run, and provides utility functions to application programs so that they don't have to do everything themselves.

About 95% of all desktop client computers run an operating system called Windows, sold by Microsoft Corporation. Virtually all the rest run Mac OS, sold by Apple Computer, Inc. Because the operating system determines so many of the distinctive characteristics of a computer, giving it its “personality,” there's a tendency to associate operating systems with hardware, and so people speak of a PC and a Windows computer interchangeably, even though the PC is the computer hardware, whereas Windows is the operating system software.

The operating systems used on ordinary client computers are not generally well suited to use on Web servers, because Web servers have very different requirements from those of desktop computers. There are several versions of Windows, and some versions (such as Windows Server 2003) can be used to run Web servers; likewise, there are several versions of Mac OS, and one of them (Mac OS X) can be used to run Web servers. However, most Web servers run an operating system called UNIX, or some other operating system closely related to UNIX, such as Linux distributions. (Mac OS X is closely related to UNIX, but it has been heavily modified to suit the desktop environment of a personal computer.)

UNIX or one of its ilk is usually chosen for Web servers because it was originally designed for use in server computers and related environments. It is an operating system that functions in the traditional mold of large-scale time-sharing systems, which were used widely before personal computers came along and which are still used very extensively today, except on the desktop. UNIX provides many features that are ideally suited to Web servers and other types of Internet servers, and indeed the Internet was largely built around UNIX, or *vice versa* (these days it's hard to separate the histories of the two).

Running a Web server generally requires a lot of technical expertise, and it requires even more when the operating system is UNIX. Virtually all large Web sites are hosted on servers running some type of UNIX or a related operating system. The most popular operating systems in this category is FreeBSD and Solaris, followed by various Linux distributions, all of which are UNIX-like. The only desktop-style operating system that has a significant presence in the world of Web servers is Windows Server 200x. It is worth noting that some UNIX-like operating systems are open-source and free, including FreeBSD and some versions of Linux.

### 6.3.2 Web server programs

A Web server must run a special application program called a Web server program in order to host a Web site. The

server program “listens” for incoming requests from the Internet and responds to them by finding the appropriate Web-site content and forwarding it to the requesting client computers. Like the Web server itself, the Web server program must run continuously, 24 hours a day and seven days a week.

The most popular Web server program is Apache, an open-source application designed to run under UNIX and its brethren, although it will also run under some versions of Windows and Mac OS X. In a distant second place is Microsoft’s Internet Information Server, which runs only under certain versions of Windows.

Like Web servers and operating systems, Web server programs require considerable technical expertise to administer, and are not for the faint of heart.

### 6.3.1 Supporting Software

Some Web-authoring tools require the installation of special software on the Web server. A typical example is Microsoft’s FrontPage server extensions, which are required for use of older versions of FrontPage and in certain other situations.

Adding features like PHP server-side scripting, Perl support, SSL, database support, and others can require the installation of additional software packages in addition to or combined with Web server programs or the operating system.

Some standard functions, such as FTP support for transferring files to and from the Web server for site construction and maintenance, are provided with most operating systems, particularly in the UNIX family. Standard versions of the support programs for these functions can be replaced with more functional open-source or commercial versions if needed or desired, in most cases.

### 6.3.2 Other Functions of the Server

A server computer used to host a Web site can also be used to handle other domain-related or organizational server functions, such as e-mail traffic, DNS, time synchronization, and so on. A discussion of these functions is beyond the scope of this paper. Suffice it to say that it is possible to use a server computer for more than one server-related purpose, if the server capacity is sufficient and if this be the desire of the webmaster or hostmaster. Thus, if you host your own Web site on your own computer, you can generally use the same computer to handle all of the e-mail for your domain. Obviously, the choice depends on how much work you are willing to do in maintenance and administration of the server, how large the server is and how heavy the load from traffic on the site happens to be, and so on.

## 6.4 Security

Unlike client computers, which can be secured in such a way that they do not respond at all to incoming, unsolicited traffic from the Internet, server computers cannot be completely isolated, because they must answer requests from other computers on the network. This means that securing Web servers is much more difficult and delicate than securing a client computer.

If you choose external hosting for your Web site, many security aspects may be handled by the Web-hosting company. However, as you move up to more elaborate hosting plans, more and more responsibility for system security will fall upon you, requiring more and more technical sophistication on your part. And if you are hosting your own site with your own computer, all security responsibility falls upon you, and extreme prudence is required in setting up and operating the Web server.

### 6.4.1 Firewalls, Routers, and NAT

A *firewall* is a hardware or software device that prevents unwanted and/or dangerous traffic from passing between two networks, such as a local area network and the Internet.

If you are operating your own Web server, in most cases it will be on your own LAN, along with your client computer and whatever other computers you might have. It’s important to have a firewall to isolate your LAN from the outside Internet. Typically, the firewall is a separate hardware device that sits between the LAN and the equipment that provides your Internet connection (ADSL or cable broadband connection). Sometimes the firewall is incorporated into a *broadband router* that can both route traffic to and from the Internet connection for your LAN and filter the traffic to remove anything that is unwanted.

The firewall is used to forbid certain traffic based on specific criteria, such as IP addresses, protocols, or port numbers. In a typical configuration, you might set up your firewall to reject all incoming, unsolicited Internet traffic to all ports except port 80 (the HTTP port used by your Web server). If you use your server to handle both your Web site and your e-mail, you’d normally open ports 80 and 25, the latter being the SMTP port used for incoming e-mail.

Combined firewall/routers usually provide *network address translation* (NAT) as well, multiplexing the traffic from several machines on your LAN with separate, private IP addresses onto a single Internet connection with a single IP address. NAT changes the source port numbers of outbound traffic to avoid conflicts when combining outbound

traffic from several machines, then uses these same ports to reroute the traffic to individual machines on the LAN when inbound traffic arrives. Normally you will set the NAT configuration to direct all unsolicited incoming traffic to your server, and you will set your firewall so that the only traffic allowed to reach your server is expected traffic such as Web requests on port 80. You'll also review the configuration of your server to make sure that other ports that are not needed are kept closed.

Sometimes you can run specific ports open on the server, but block them with the firewall, making the services on these ports available only to other machines on your LAN. For example, you could run a POP server (for e-mail) or an FTP server on your server and block the corresponding ports on your firewall, so that the only machines that could connect to FTP or POP services on your server would be those on your own LAN.

Some operating systems provide or support software firewalls that run right on the computer itself. These are generally inferior to independent hardware firewalls and should be avoided, unless you have nothing else between your machines and the Net (*i.e.*, they are better than nothing, but not by much).

If this all sounds complicated ... it is. It's all part of what you must deal with if you choose to run your own Web server.

### 6.4.2 Viruses, Trojans, and Other Malware

The situation for viruses and other malware on servers is somewhat different from that on client computers, provided the servers are not running standard desktop operating systems. There are very few instances of malware that afflict UNIX servers, so the overall threat is lower; but on the other hand, servers must be at least partially exposed to the Internet, making them easier targets for attack. Adware and spyware are practically nonexistent on UNIX servers, since these are not client machines.

There are few antivirus products for UNIX systems, and even where antivirus products exist (as on Windows systems), they often cause more problems than they solve. Protecting against malware consists of prudent operating procedures and vigilance. Trojan horses, worms, and the like often profit from bugs in various *daemons* (a daemon is simply a server program under UNIX—under Windows they are called *services*) that listen to open ports. It's important to keep the operating system and the daemons up to date so that any security bugs that may exist in them are closed before outside hackers can exploit them.

### 6.4.3 Backups

Backups are a good idea for all computers, everywhere. They are especially important for servers that hold large amounts of important information.

A backup is simply a procedure that copies information from a computer to some other place for safekeeping, so that the information can be restored in the event of a hardware or software failure on the computer. Backups also provide a way to restore information that you may have accidentally deleted or modified unintentionally. There are many different ways to carry out backups; how they are done is not very important, as long as they are done.

A common way to backup a server (or any other type of computer) is to copy the contents of one disk to another, if there are two or more disks in the machine. Special programs exist that can facilitate this operation, either included with the operating system (such as the `tar` command on UNIX or the backup utility included with some versions of Windows) or obtainable from third parties (such as Norton Ghost or Acronis TrueImage). Servers are also backed up to tape drives in some cases, although tape drives are expensive.

Backing up data to other disks in the same machine guards against disk failure. To guard against total loss of the machine, the backup must target media that are physically separate from the machine. For example, backing up to tape allows the tape to be put in a safe place, away from the server, thereby ensuring that the tape will not be lost even if the server is destroyed.

It is also possible to perform backups by copying data to recordable CDs or DVDs or other media. In the case of write-once media such as CD-Rs, the backups can double as archives.

The frequency of backups is up to you. Daily backups may be justified if the data on a computer changes frequently; weekly backups may suffice for a Web server that merely hosts a static Web site that is rarely changed. Even less frequent backups may be enough for very stable and static servers. Conversely, e-commerce sites may require several backups per day.

In most cases, backups can be carried out even while the system is running, but it depends on the procedure and software used for the backup.

If your Web site is externally hosted, the Web-hosting company will normally take backups of your site regularly. If their server is destroyed or other problems ensue, they will restore your Web site from their backups. If you accidentally destroy information on your server, you can ask them to restore it from their backups, but they will usually charge for this service.

## 7 Promoting Your Web Site

If you publish a Web site, it's reasonable to assume that you'd like people to visit it. In order to get visitors, you need to make your site known, and in some cases you may also wish to actively promote it as well.

### 7.1 Out-of-Band Advertising and Promotion

Out-of-band means "outside the world of computers," and so out-of-band advertising and promotion simply means making your Web site known and/or promoting it by means other than via a computer.

For example, the most common way to make your site known in this way is to print the URL of your site on business cards, letterheads, on your products or product documentation, or in print advertisements that you submit to newspapers and magazines. Visitors then type your URL manually into their browsers in order to reach your site.

Unless you advertise very actively in this way, relatively few visitors are likely to reach your site as a result of out-of-band advertising or promotion. Of course, if you take out full-page advertisements in the *New York Times*, you might get quite a bit of traffic as a result. Out-of-band techniques are usually most useful for reinforcing contacts you've already made; for instance, by giving someone a business card with your URL on it, you allow him to learn more about your products and services by visiting your site, and so his contact with you is not limited to just the in-person meeting.

A special case of this type of advertising exists when the news media mention your site and provide the URL. This can provide a huge jump in traffic to your site, sometimes enough to crash Web servers or clog your Internet connections. The gain in awareness (and possibly in business, if you are running a business Web site) may or may not compensate for the severe overload of your server and connection. If your site is being externally hosted, you do risk being charged for extra bandwidth, or having your site shut off by the Web-hosting company if it generates enough traffic to interfere with their other customers or overload their network.

### 7.2 General Linking

A common way for people to reach a Web site is via links to that site on other sites. Anyone can link to your Web site, and depending on how much traffic *they* receive on their site, this may or may not drive significant traffic to your site.

Links from personal Web sites typically generate very little traffic. The more popular the site linking to yours, the more traffic you will receive from it. If the site of a large news media organization links to your site, you may see a tidal wave of traffic such as we've already described above, which can be either good or bad, depending on circumstances. Links from sites that have highly time-dependent content (such as news sites) tend to generate an initial avalanche of traffic that rapidly tapers off until it merges again into the background noise.

Many small sites participate in *link exchanges* or *rings* in which individual sites with some commonality of content (all needlepoint sites, for example) agree to link to each other to drive traffic to all of the sites. Since this provides an easy way for interested visitors to visit multiple sites on the same topic, it can increase site traffic for everyone in the exchange or ring. Initially very popular, these arrangements are less common than they once were. They have always been favored primarily by very small sites and personal sites that don't have other, more effective ways to drive traffic. Their slight decline has been driven in part by the rise in importance of the search engines.

### 7.3 Search Engines

Search engines are Web sites that compile gigantic indexes of other sites on the Web, and then allow visitors to search these indexes using keywords, in order to find Web sites relevant to their interests. Search engines have evolved into the leading sources of traffic for most Web sites today. Chances are that most of the traffic to your own Web site will be driven by search engines, unless you specifically exclude your site from the search-engine indexes.

Search engines produce a list of links to sites that match the keywords entered by visitors. Visitors by nature tend to click on the links that appear first on the list, particularly those that appear on the first page of the list. Thus, the higher a site appears on the list, the more traffic it receives.

#### 7.3.1 Placing Your Site in a Search Engine

Placing your Web site in the index of a search engine is easy. You simply visit the Web site of the search engine and find the instructions on how to submit the URL of your site (instructions are always available somewhere, often in tiny print or via a small link somewhere on the search-engine page). Once you add your URL, the spiders and crawlers (robotic programs that continuously scour the entire Web) of the search engine will begin to visit your site and add your pages to the index of the search engine. Within a few hours, days, or weeks, your site will begin

appearing in the index of the search engine, and people searching for Web sites with keywords relevant to your site will see a link to your site displayed in the results provided by the search engine. They can then visit your site by clicking on the link.

The exact methods by which search engines decide the order in which they list sites for a given combination of keyboards are usually closely guarded secrets—mainly because the search-engine companies do not want spammers and other troublemakers to try to skew the search results by taking advantage of any idiosyncrasy of the search-engine algorithms. Nevertheless, most search engines publish general information on how they determine their site rankings, and typically their methods are based on automated, complex analysis of the actual content of a Web page, especially the text. Since the reputations of most search engines depend almost entirely on their claimed ability to remain fully objective, they work hard to avoid anything that might cause inappropriate sites to appear high on their lists for a given set of keywords. For example, if you search for “engine oil change,” search engines will try to point to sites that truly are relevant to the changing of oil in engines, and not sites that refer to locomotives, petroleum exploration, or other less relevant topics. Search engines also are engaged in a continuing and bitter battle with spammers and others who try to trick their robots into listing certain sites higher than they deserve to be listed. This is especially common among owners of sites that are a bit shady to begin with, such as pornographic and scam sites.

Search engines also try to track the number of people who visit specific sites through their search engines, and they incorporate this into determination of their rankings. The more popular a site becomes, the higher it ranks. Spammers and bad people try to trick this system as well, and search engines must work hard to stay ahead of the con artists.

### 7.3.2 Optimizing Your Placement

By far the best way to ensure that you will be appropriately ranked by search engines is to include plenty of coherent content on your site, especially in the form of text. Modern search engines evaluate sites in very complex ways designed to take into account all the content on a site. Trying to skew the evaluations of the search engines by filling a site with specific keywords or by other methods is both counterproductive (because it drives the wrong kind of visitors to your site) and very irritating to the search-engine operators. In the worst cases, the administrators of search engines may permanently remove your site from their indexes if it becomes obvious that you are trying to change your *page rank* (the rank of your Web site on their listings) through fraudulent or dubious methods.

Search engines are not currently intelligent enough to properly evaluate images on sites. If your site is mostly images (a photo album, for example) and you want search engines to index it, you should include some descriptive text along with your images so that search engines can index the site properly. In general, the more text, the better.

When you first submit your URL to a search engine, you should not expect to see yourself immediately appear high in the rankings. Over time, if your site is popular, its popularity will drive it higher in the search-engine ranks, and the higher ranks will in turn drive more traffic to your site.

## 7.4 Rating Systems and Filters

Many people believe that children (and sometimes adults) must be protected from certain types of online content, such as violence, pornography, unpopular opinions, and so on. Various mechanisms have evolved on the Web to facilitate censorship, for better or for worse, and in some cases they may affect your Web site.

*Rating systems* are mechanisms that allow Web sites to obtain or create ratings for themselves that reflect the type of objectionable content they contain (if any). Browsers that are capable of recognizing the ratings then prohibit access to sites that contain unacceptable ratings, with the threshold of unacceptability being set by the owner of the client computer. The threshold settings are protected by a password or other means, so that once they are set, anyone who does not have the password can see only Web content that falls within the range of acceptable ratings. The Internet Content Rating Association (ICRA) is a prominent player in the field of Web site ratings.

*Filters* are programs or add-ons to browsers that prohibit access to certain Web sites or censor certain sites or certain types of content. Programs such as CYBERSitter, Net Nanny, and Cyber Patrol prevent a browser from accessing certain sites on internal *blacklists*, and also may block sites based on the type of text they contain, or may edit the content they allow to be displayed in order to remove offensive text. Most vendors of such software publish neither their blacklists nor the criteria they use for choosing which sites or content to block. Some filters are incorporated upstream of the client computer, that is, they may be operated by the ISP.

Filters and rating systems on the client computer can be circumvented by clever users in most cases. Those operated by the ISP usually cannot.

The ostensible purpose of rating systems and filters is to protect children from inappropriate content, but some countries and organizations use such systems to prevent adults from seeing certain types of content as well.

If your site is on a blacklist used by a filter, some visitors may not be able to access your site or view it as you intended. If your site has an objectionable rating, some visitors may be unable to see the content with the objectionable rating. In some cases, this may be your own doing, as when you self-rate your site using ICRA guidelines. In other

cases (most cases), you may have no control over it, as when you are blacklisted by a filter vendor or by an ISP that operates a filter.

If your site contains content that may upset many visitors, it's not a bad idea to put a disclaimer on the home page of the site. This may also help prevent you from being sued, in some cases.

## 7.5 The Need for Traffic

There's no law that says you must have lots of traffic to your site. Indeed, heavy traffic tends to overload servers and Internet connections, requiring you to upgrade your hardware or pay for a faster Internet connection. If you are externally hosting your site, you may run up huge supplemental charges if heavy traffic overloads the Web-hosting company's server, or if you exceed your allotted bandwidth.

There is a certain psychological satisfaction to receiving lots of visitors on your Web site; it's up to you to decide if the satisfaction is sufficient to compensate for the potential costs. More significantly, if you are running a business site, there may be a direct link between how much traffic you receive and how much business you do, particularly if your site is an e-commerce site from which visitors can purchase your products or services directly. In this latter case, the cost of handling more traffic to the site is almost always more than eclipsed by the increased revenue generated by more visitors.

Note that many sites receive very modest traffic, so much so that almost any computer and any type of Internet connection is more than sufficient to handle the traffic, so the cost of heavy traffic won't necessarily ever become an issue.

## 8 Threats to Your Site

Webmasters occasionally encounter problems of various types while running a Web site. In this section, we explore some of the possible issues. Nothing here is intended to replace advice from a legal professional.

### 8.1 Copyright Infringement

Anything published on a Web site is protected by copyright, as a general rule. Anything you create yourself to publish on your Web site is protected by copyright, and the copyright is owned exclusively by you. Similarly, any content you see on any Web site anywhere is generally protected by copyright, also, and the copyrights for various works belong to their respective authors. You don't have to take any special action to enjoy copyright protection; the mere fact of creating content for your site causes it to be automatically protected by copyright, and as the creator of that content you are the only person who can authorize its use. Copyright protection generally lasts for your entire lifetime, plus seventy years (depending on the jurisdiction).

The law has not kept pace with technology, and there are some gray areas today. However, it is safe to say generally that nobody can legally use the content you create for publication on your own Web site without your authorization, and conversely, you cannot use the content of other sites you see on the Web without obtaining the authorization of whoever holds the copyright(s) on that content. If someone uses your content without your authorization, you can sue that person, or even file a criminal complaint (in some jurisdictions). And if you use someone else's content without that person's authorization, he or she may sue you, and may even file a criminal complaint against you.

All types of creative content are covered by copyright, including images (drawings, photos, etc.), text, music, sound effects, and so on. Content that has absolutely no creative element in it at all, such as simple facts (the height of Mount Everest, or the population of Malta), is not and cannot be protected by copyright.

At least that's what the law says, in most jurisdictions.

#### 8.1.1 Infringements on Your Copyrights

When someone copies content from your site without your permission, for virtually any type of use, she is infringing on your copyright, and you can take legal action against her. Doing this in theory, and doing this in fact are, however, two very different things.

You can take for granted that anything you publish on your site can and will be stolen by others. It doesn't matter that it is illegal to do so; it will be stolen, anyway. Never put any content on your site that you do not wish to have stolen. If you've spent the last ten years of your life writing a novel, and you don't want it stolen, do not publish it on your Web site; if you wish to make people aware of it, publish only a few excerpts to give them a feel for the novel (and remember that these excerpts will be stolen).

Similarly, if you put images on your Web site, accept that they will be stolen. If you are publishing photos that you

wish to sell or license, make sure that the versions on the Web site are of poor enough quality that you are willing to have them stolen. Keep the high-quality versions off the site, and make people pay you to obtain high-quality versions.

In the case of dramatic and severe infringement, it may be worth your while to take legal action. Recognize, though, that legal action costs a lot of money, and unless you are wealthy and you wish to pursue the matter purely out of principle, many infringements may cost you less than the legal fees would be to take action against the infringers. Additionally, if the infringer is in a different legal jurisdiction, especially one without strong laws protecting copyrights, you may be completely out of luck, and there is nothing you can do. This is another reason why you should not publish anything that you don't want stolen.

In some jurisdictions, some types of unauthorized use of copyrighted materials are permitted by law. For example, use of portions of your site content for educational purposes (as examples shown in class, for instance) is legal in many jurisdictions, including the United States) the United Kingdom, and France. Parodies are also legal in many jurisdictions. Some jurisdictions permit private copies of your content to be made without your explicit authorization.

### 8.1.2 Registering Your Copyrights

As a general rule, anything you create is covered by copyright—you don't have to take any special action to protect it. However, in some jurisdictions, officially registering your copyright will allow you to obtain larger damages in court should you decide to sue someone, and will make it easier for you to prove that something is yours.

In the United States, for example, you can register your copyrighted works with the Library of Congress to obtain additional protection and simplify pursuit of infringers. France provides a *dépôt légal* for the same purpose. In the United Kingdom, the British Library registers copyrighted works.

In the United States, if you see your copyrighted work appearing without authorization on another Web site, you can file a DMCA take-down request to have the work removed from the site, if you are willing to follow through with a lawsuit and accept United States jurisdiction for your claim.

### 8.1.3 Respecting Other Copyrights

Just as others must respect your copyright on the content you create yourself for your site, legally you must respect the copyrights of others on the content published on their sites. If you copy material from other sites without authorization from the copyright holders, you risk a lawsuit or even criminal prosecution, depending on your jurisdiction and that of the copyright holder. Publishing content on the Web does *not* release it to the public domain.

In the United States, some organizations may “pull a DMCA” against your site, using specific details of United States copyright law to harass you or effectively censor your site. The Digital Millennium Copyright Act (DMCA) provides for copyright holders to place Web sites on notice that they are infringing on copyrights. When this notice is given, the potentially infringing material must be immediately removed by any provider acting to host the Web site (ISP or Web-hosting company), or that provider risks legal action itself. Most providers will take the site down to protect themselves legally when they receive a notice of this kind, even before the entity filing the complaint proves that any actual infringement has occurred. This can take a site down for two weeks or more. If the entity making the notice does not follow through, it becomes moot, but it's still a way to take a site down for days or weeks, which can effectively cripple some organizations with Web sites. Organizations use the DMCA to try to censor content with which they disagree; for example, corporations will file DMCA notices in order to try to remove Web site content that is critical of them.

## 8.2 Trademark Infringement

Trademark infringement may occur if you use a trademark owned by some other entity on your Web site in a way which damages the owner of the trademark—by confusing consumers, or causing loss of goodwill towards the trademark owner, or diluting or corrupting the significance of the trademark, and so on.

The best way to avoid trademark infringement is to avoid the use of trademarks anywhere on your site, particularly logo trademarks (trademarks that are visual in nature, as opposed to a trademarked word or phrase).

## 8.3 Libel

Libel is written defamation. It's possible, though not probable, that someone might sue you for libel if you are harshly critical of a person or organization, particularly if there is evidence that your words have damaged that person or organization in some tangible way (loss of business, loss of reputation). The exact criteria vary from one jurisdiction to another.

In this context, Web publication is very much like print publication, so similar rules for libel apply. This is still a bit of a legal gray area, however. In some countries, freedom of speech is strong enough to counter many accusations of libel; in other countries, claims of libel tend to have more weight than arguments of freedom of speech. Merely

hurting someone's feelings doesn't usually qualify as libel in most jurisdictions, but in some jurisdictions it might.

## 8.4 Prohibited Content

There are very few places in the world where you can get away with putting absolutely anything on a Web site. Even in countries that claim freedom of speech, some types of content are prohibited. Depending again on your jurisdiction, prohibited content may include religious materials, some or all types of pornography (and the definition of pornography itself varies widely), political commentary or activism, depositories of copyrighted material that isn't your own ("warez" sites that provide pirated software, etc.), certain types of software or technical information (methods for defeating copy protection, encryption software, weaponmaking cookbooks, etc.), and so on.

Sometimes prohibited content is simply pulled off a site. Sometimes the webmaster is executed. Other cases fall somewhere in between. It all depends on the content and the jurisdiction. What is tolerated without a problem in one country might be a death sentence in another. Be careful where you host your site, and where you live, as both jurisdictions may have claims.

## 8.5 Attacks Against Your Site

Some individuals may try to attack your Web site in technical ways, by profiting from bugs in the software on your Web server or errors in the configuration or construction of your Web site. The usual motivation is pure vandalism or (increasingly) a desire to take over the computer holding your Web site in order to use it for illegitimate purposes (forwarding of spam e-mail, hosting of pirated or other illegal materials, launching of coordinated attacks against other computers, etc.).

If your Web site is hosted by a third party, usually the third party will take some steps to keep the Web server secure. If you host your own site, it's all up to you. In some cases, things that you do with your site can make it more vulnerable to attack.

### 8.5.1 Viruses

*Virus* infections of Web servers are rare, simply because there are very few computer viruses in the wild that are designed to attack UNIX servers and Web server programs. Additionally, there are few vectors for virus infection on Web servers, since they are not normally used to surf the Web or to open e-mail messages and attachments.

Web servers running under Windows may be vulnerable if the machines are also used as clients. This is one reason why it's a very bad idea to use the same computer as both a client and server.

Antivirus products for Web servers are few and far between, and are usually not a good idea, as the risk is very low, and antivirus software can cause more problems than it solves, especially on production servers.

### 8.5.2 Worms

*Worms* are similar to viruses, except that they don't require any explicit action on the part of a human being in order to infect a machine. Because they can infect machines automatically, worms are much more of a danger for Web servers than viruses.

Web servers are typically infected by worms via the Web server program itself, through exploitation of a bug in the server software or an error in the server configuration. The use of server-side scripts and CGI can open the door to worms if the scripts and CGI programs are not very carefully written.

It's important to keep the software on a Web server up to date and appropriately patched for any outstanding security issues. This prevents worms from entering the server through flaws in the software.

Scripts and CGI programs must be vetted with extreme care to ensure that they provide absolutely no opportunity to enter the machine or to do anything other than what they've been written to accomplish. A classic vector for worm infections is a buffer overflow in a CGI program.

The configuration of the Web server must also be carefully reviewed to ensure that it provides no opportunity for worms to enter the system. If the Web server includes add-ons to interface with client-side products such as Web-authoring tools, these add-ons must also be kept up to date and carefully reviewed to make sure that they do not create any security holes. The misuse of the proprietary interfaces provided for Web-authoring products is one of the most common ports of entry for worms.

Web servers should be dedicated to hosting Web sites, and nothing else, if possible. Incoming ports should be blocked by independent firewalls for all port numbers except the ones required by the server (typically only port 80 for Web servers). No software other than the Web server software should be running on the machine, and the machine should not be listening on any ports other than the Web server ports. Note that Windows machines are much more difficult to secure in this way than UNIX machines, because so much additional software is constantly running on Windows machines, and some of it continuously listens on specific ports, whether you want it to or not.

It can be difficult to detect worm infection. The server must be carefully watched on a regular basis, and any unex-

plained changes noted and investigated. Not only should incoming ports be filtered by an upstream firewall, but outbound connections to unusual ports should be blocked and logged as well. The appearance of unusual files on the server, or unusual changes in existing files, must be investigated. Unusual activity on the server must also be examined.

If a single server is used for multiple purposes, such as Web hosting, e-mail management, and so on, then much greater vigilance is required, since the opportunities for worm infection are greatly expanded. Running UNIX instead of Windows on the server can make the machine much less vulnerable. If the server is UNIX, it should be run without X servers or clients (the X environment on UNIX provides a Windows-like graphic user interface—which isn't really necessary for a machine that will never be used as a client).

Various operating systems provide for tight lockdown of the system for specific purposes. This type of lockdown is a good idea for computers that are acting as servers.

Worms differ from viruses in that worms can propagate from one computer to another without any human interaction.

### 8.5.3 Trojan Horses

A *Trojan horse* is a computer program that seems innocent and useful, but in fact hides a secret and malevolent purpose. When it is executed by a user with a high level of privilege for its apparent legitimate purpose, it also carries out other, secret, dangerous operations of which the user is unaware, compromising the security of the computer system.

Since Trojan horses infect a computer through the execution of an untrustworthy program, they are not common on Web servers. Most Web servers execute a handful of highly trustworthy programs continuously in a relatively stable environment. However, if a system administrator executes an untrustworthy utility program on the server, a Trojan horse infection may be the result.

The best defense against Trojan horses is to never execute an untrustworthy program on a Web server under a privileged user account. Here again, the UNIX environment has an advantage because many programs are available in source form and can be recompiled from source to help exclude Trojan horses.

Trojan horses can be extremely difficult to detect. The methods are similar to those mentioned for worms, above. If changes in a system are noted not longer after the use of a new program by a privileged user, this may be a sign of a Trojan horse infection.

Trojan horses differ from viruses in that Trojans do not infect other programs with copies of themselves. Trojans differ from worms in that they require human interaction in order to spread infection.

### 8.5.4 Denial-of-Service (DoS) Attacks

Most Web servers keep logs of all activity occurring on the system. If you look at the logs for your Web server (which you will have if you run your own Web server, or if you have an advanced plan with a Web-hosting company), you will invariably see evidence of constant attacks against it. Most are ignored by the server or fail to work as the attacker hopes, and so you need not be concerned about them.

Sometimes, hackers on the Internet will try to overload your Web server in order to make it unable to respond to legitimate requests. This is called a *denial-of-service (DoS)* attack. A DoS attack does not necessarily crash your Web server, nor does it prevent it from operating, but it does put a load on the machine that is so heavy that normal requests from visitors to your Web site are answered only after seconds or minutes, or perhaps not at all (compared to the fraction of a second required for a response under normal conditions).

Denial-of-service attacks are not random; they are the result of some party deliberately targeting your Web site for the attack. Severe attacks can overload not only the server but also its connection to the Internet. Most attacks are motivated by someone's disagreement with something you say on your site, or, in commercial environments, by the desire of a dishonest competitor to prevent you from operating your e-commerce site normally. Some attacks are politically motivated.

Since DoS attacks are deliberate and targeted, in some jurisdictions they come under the legal heading of terrorism. There isn't any easy way to stop a DoS attack short of taking a Web server offline (which only accomplishes the attackers purpose). In some cases the source of the attack can be identified by very careful detective work—this requires an expert in this type of investigation, however. Fortunately, DoS attacks are rare, unless you have an extraordinarily controversial or high-profile Web site.

# HTML Quick Reference

`<html> ... </html>`

These tags enclose the entire Web page. Most browsers do not require them, but the official HTML standard does. All of the other tags shown on this page should always appear somewhere between the start and end html tags (that is, the first tag on a Web page should be `<html>`, and the last should be `</html>`).

`<head> ... </head>`

These tags enclose header information internal to the page, if any. Of the tags shown below, only the title tag appears between head tags; all others appear between the body tags. The area enclosed by head tags must precede the area enclosed by body tags, and they must not overlap.

`<title> ... </title>`

The text between these tags is displayed on the title bar of the browser window (usually) when the page is being displayed. This pair of tags should always appear between head tags.

`<body> ... </body>`

The body tag encloses the actual body of the page (the part that is to be rendered in the browser window). This pair of tags only occurs once in a Web page. All of the remaining tags shown below should always appear somewhere between the start and end body tags.

`<p> ... </p>`

These tags enclose a paragraph. The browser introduces a line break before each paragraph, and separates paragraphs by a small space. The ending tag is optional.

`<br>`

This tag introduces a line break in the formatted text. Without this tag, the browser will run all text in a paragraph together into one block, no matter how it appears in the actual Web-page file.

`<hn> ... </hn>`

The `hn` tags enclose the text of a heading. Most browsers render headings in bold and in a larger font than that of running text. The *n* can be from 1 to 6 (h1, h2, h3, etc.). The h1 heading is the top level, the h2 heading is the next subheading below that, and so on.

`<b> ... </b>`

All text between these tags is displayed in **boldface**.

`<i> ... </i>`

All text between these tags is displayed in *italics*.

`<u> ... </u>`

All text between these tags is displayed underlined.

`<a href="(url)"> ... </a>`

These tags enclose text or content that is to be a hyperlink. The portion shown as *(url)* should be replaced with the URL of the page or other resource to which the hyperlink is to point. This is the normal way to include links to other pages on a Web page.

``

This tag identifies an image file or other resource that is to be inserted into the current page at the spot where the tag is located. The portion shown as *(url)* should be replaced with the URL of the image or resource to be inserted.

*A complete HTML reference is available at*

<http://www.w3.org/TR/1999/REC-html401-19991224>





