

# LESSON 1

# What Is Web Publishing?

A journey of a thousand miles begins with a single step, and here you are in Lesson 1 of a journey that will show you how to write, design, and publish pages on the World Wide Web. But before beginning the actual journey, you should start simple, with the basics. You'll learn the following:

- How the World Wide Web really works
- What web browsers do, and which browsers your audience will be using
- What a web server is, and why you need one
- Some information about *uniform resource locators* (URLs)

These days, the Web is pervasive, and today's lesson might seem like old news. If so, feel free to skim it and skip ahead to Lesson 2, "Getting Your Tools in Order," where you'll discover the first steps you need to take to learn to create web pages.

## Thinking Like a Web Publisher

You're almost certainly already familiar with the Web as a user. You open your favorite web browser and visit websites where you look up information, shop, or keep up with what your friends are doing. You may also use your web browser to read your email, check your calendar, and do your work.

Being a web publisher means understanding what happens when you enter an address in your web browser or click a link and visit a website. But first, before I get into explaining the Web at a technical level, I want to define it at a conceptual level.

The Web is

- A hypertext information system
- Cross-platform
- Distributed
- Dynamic
- Interactive

So, let's look at all these words and see what they mean in the context of how you use the Web as a publishing medium.

### The Web Is a Hypertext Information System

The idea behind hypertext is that instead of reading text in a rigid, linear structure (such as a book), you can skip easily from one point to another. You can get more information, go back, jump to other topics, and navigate through the text based on what interests you at the time.

*Hypertext* enables you to read and navigate text and visual information in a nonlinear way, based on what you want to know next.

When you hear the term *hypertext*, think *links*. (In fact, some people still refer to links as hyperlinks.) Whenever you visit a web page, you're almost certain to see links throughout the page. Some of the links might point to locations within that same page, others to pages on the same site, and still others might point to pages on other sites. Hypertext was an old concept when the Web was invented—it was found in applications such as HyperCard and various help systems. However, the World Wide Web redefined how large a hypertext system could be. Even large websites were hypertext systems of a scale not before seen, and when you take into account that it's no more difficult to link to a document on a server in Australia from a server in the United States than it is to link to a document stored in the same directory, the scope of the Web becomes truly staggering.

**NOTE**

Nearly all large corporations and medium-sized businesses and organizations are using web technology to manage projects, order materials, and distribute company information in a paperless environment. By locating their documents on a private, secure web server called an *intranet*, they take advantage of the technologies the World Wide Web has to offer while keeping the information contained within the company.

## The Web Is Cross-Platform

If you can access the Internet, you can access the World Wide Web, regardless of whether you're working on a smartphone, a tablet, a brand new laptop, or a desktop computer you bought at the flea market. If you think Windows menus and buttons look better than Macintosh menus and buttons or vice versa (or if you think both Macintosh and Windows people are weenies), it doesn't matter. The World Wide Web isn't limited to any one kind of machine or developed by any one company. The Web is entirely cross-platform.

*Cross-platform* means that you can access web information equally well from any computer hardware running any operating system using any display.

### The Cross-Platform Ideal

The whole idea that the Web is—and should be—cross-platform is strongly held to by purists. The reality, however, is somewhat different. With the introduction over the years of numerous special features, technologies, and media types, the cross-platform nature of the Web has been compromised. Web authors can choose to use nonstandard features, like Flash, but in doing so they limit the potential audience for their site, especially as more and more people switch to smartphones and mobile devices to view the Web. Web publishers also must choose between creating native applications for mobile devices or using modern web standards to build web applications that are more cross-platform compatible. It's up to individual creators to decide whether to compromise cross-platform flexibility for the greater capabilities of a proprietary platform.

## The Web Is Distributed

Web content can take up a great deal of storage, particularly when you include images, audio, and video. To store all the information published on the Web, you would need

an untold amount of disk space, and managing it would be almost impossible. (Not that there aren't people who try.)

The Web succeeds at providing so much information because that information is distributed globally across millions of websites, each of which contributes the space for the information it publishes. These sites reside on one or more computers, referred to as web servers. **A web server is just a computer that listens for requests from web browsers and responds to that request.** You, as a consumer of that information, request a resource from the server to view it. You don't have to install it or do anything other than point your browser at that site.

A *website* is a location on the Web that publishes some kind of information. When you view a web page, your browser connects to that website to get that information.

Each website, and each page or bit of information on that site, has a unique address. This address is called a *uniform resource locator* or **URL**. When people tell you to visit a site at <http://www.nytimes.com/>, they've just given you a URL. Whenever you use a browser to visit a website, you get there using a URL. You'll learn more about URLs later in this lesson in the "Uniform Resource Locators" section.

## The Web Is Dynamic

If you want a permanent copy of some information that's stored on the Web, you have to save it locally because the content can change any time, even while you're viewing the page.

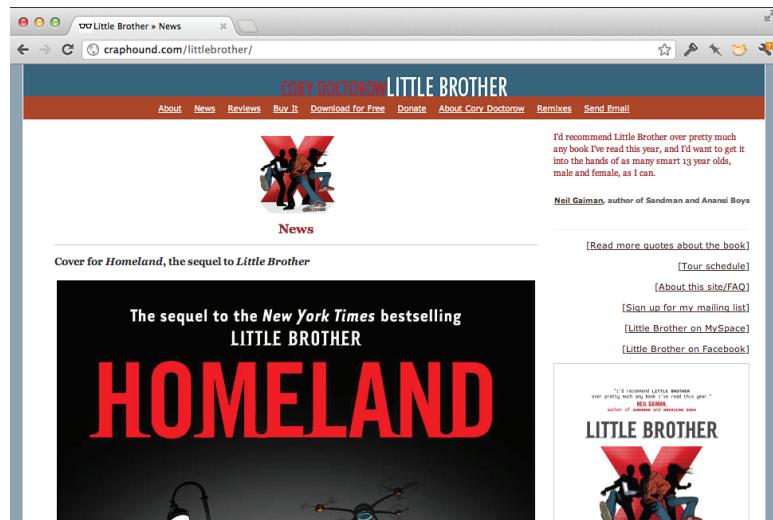
If you're browsing that information, you don't have to install a new version of the help system, buy another book, or call technical support to get updated information. Just launch your browser and check out what's there.

If you're publishing on the Web, you can make sure that your information is up-to-date all the time. You don't have to spend a lot of time re-releasing updated documents. There's no cost of materials. You don't have to get bids on numbers of copies or quality of output. Color is free. And you won't get calls from hapless customers who have a version of the book that was obsolete four years ago.

Consider a book published and distributed entirely online, such as *Little Brother* by Cory Doctorow (which you can find at <http://craphound.com/littlebrother/>). He can correct any mistakes in the book and simply upload the revised text to his website, making it instantly available to his readers. He can post pointers to foreign language translations of the book as they arrive. The website for the book appears in Figure 1.1.

**FIGURE 1.1**

The website for *Little Brother*.

**NOTE**

The pictures throughout this book are taken in Google Chrome or Safari running on OS X. The only reason for this use is that I'm writing this book on an Apple Macintosh. If you're using a different operating system, don't feel left out. As I noted earlier, the glory of the Web is that you see the same information regardless of the platform you're using.

For some sites, the capability to update the site on-the-fly, at any moment, is precisely why the site exists. Figure 1.2 shows the home page for the BBC News, a site that's updated 24 hours a day to reflect up-to-the-minute news as it happens. Because the site is up and available all the time, it has an immediacy that newspapers cannot match. Visit the BBC News at <http://www.bbc.co.uk/news/world/>.

These days, you don't even need to reload a web page to receive updated information. Through the use of JavaScript, which I discuss starting in Lesson 17, "Introducing JavaScript," you can update the contents of a page in real time. The scores and statistics on the NBA game page in Figure 1.3 are updated in place as the game progresses.

## LESSON 1: What Is Web Publishing?

**FIGURE 1.2**

The BBC News.

The screenshot shows the BBC News World homepage. At the top, there's a navigation bar with links for News, Sport, Weather, Travel, Future, TV, Radio, More..., Search, and RSS. Below the navigation is a banner for 'BBC NEWS WORLD'. The main content area features a headline: 'Liberal success in Libya election' with a sub-headline: 'Results from Libya's first elections after the overthrow of Col Gaddafi show gains for an alliance led by former interim PM Mahmoud Jibril.' Below the headline is a photo of a press conference and a sidebar with a video thumbnail of a man speaking. To the right, there's a 'Watch/Listen' section with video thumbnails for 'I nearly ate a needle in my airline food' and 'Liberal success in Libya election'. At the bottom of the page, there's an advertisement for 'audible.com'.

**FIGURE 1.3**

Live game updates on the CBS Sports website.

The screenshot shows the CBS Sports Game Tracker for the Houston Astros vs. San Diego Padres game. The top navigation bar includes links for Sign-in, Forget Log-in or Password?, Help, Not a member? Register Now!, and Stream. The main content area displays the 'DAILY LEADERS (7/17)' for the game. It shows the top players for each team in categories like AVG, HR, R, and RBI. Below the leaders are the 'Astros (34-56)' and 'Padres (36-55)' lineups. A large graphic of a baseball field highlights the pitcher, J. Lyles, and the batter, L. Forsythe. To the right, there are sections for 'Pitching Stats' (showing Lyles' stats and Marquis' stats), 'Last 2 ABs' (with notes about AB1 being grounded out and AB2 being a flyout), and 'Consumes' (an advertisement for CBS Sports Fantasy Football). The bottom of the page shows the current play at bat.

## The Web Is Interactive

**Interactivity** is the capability to “talk back” to the web server. More traditional media, such as television, isn’t interactive in the slightest; all you do is sit and watch as shows are played at you. Other than changing the channel, you don’t have much control over

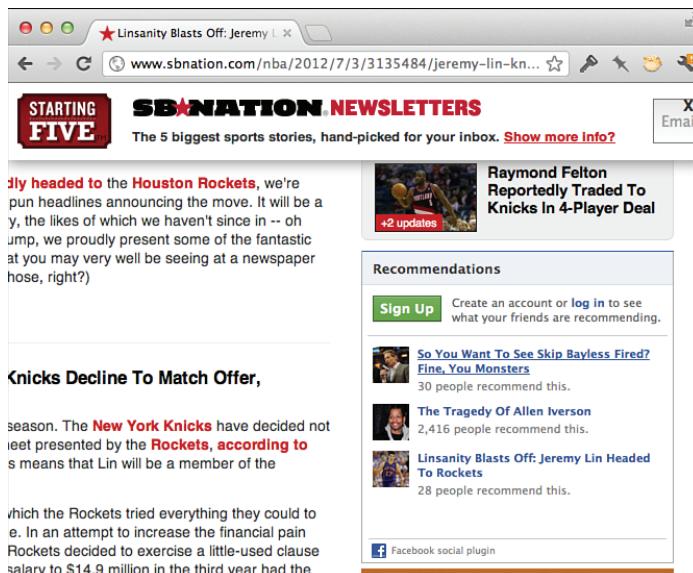
what you see. The Web is inherently interactive; the act of selecting a link and jumping to another web page to go somewhere else on the Web is a form of interactivity. In addition to this simple interactivity, however, the Web enables you to communicate with the publisher of the pages you're reading and with other readers of those pages.

Indeed, the most popular sites on the Web these days are about interacting with other users of the site rather than with the site's publisher. That's what people mean when they say "social media." Rather than spending money to hire writers and cameramen, now sites are spending money to hire programmers to create spaces for people to share content they create with one another. These days, it's not uncommon to see people on TV reading viewer posts from Twitter or Facebook out loud on the air. Such is the degree to which this form of media has taken hold.

As a web publisher, you'll need to decide the type of interaction you want your site to provide. You can publish web pages without any outlet for users to interact. You can enable users to submit feedback privately. You can enable them to publish public comments and converse with you and with each other. You can provide forums that enable users to interact with one another directly. You can provide games or other interactive features. You can even incorporate interactive features from other websites into your own so that you can integrate your site with the sites to which your users already belong. For example, Figure 1.4 shows a Facebook widget incorporated into a third-party website.

**FIGURE 1.4**

A Facebook widget.



## Web Browsers

A web browser, as mentioned earlier, is the application you use to view pages and navigate the World Wide Web. A wide array of web browsers is available for just about every platform you can imagine. Microsoft Internet Explorer, for example, is included with Windows, and Safari is included with OS X. Mozilla Firefox, Google Chrome, and Opera are all available as free downloads. Likewise, more and more people are using browsers on mobile devices and tablets. iPhone and iPad use Mobile Safari. The Android mobile platform has its own browser. There are also other mobile platforms, like BlackBerry and Windows Phone, and third-party browsers for both Android and iPhone. Not too many years ago, Internet Explorer was the dominant browser for Windows, the OS X market share was less than 5%, and mobile browsers were so limited that they wouldn't work with regular web pages at all. Back then, developers sometimes chose to support Internet Explorer and ignore other browsers. That is no longer a viable strategy.

### NOTE

Choosing to develop for a specific browser, such as Internet Explorer, is only suitable when you know a limited audience using the targeted browser software will view your website. Developing this way is a common practice in corporations implementing intranets. In these situations, it's a fair assumption that all users in the organization will use the browser supplied to them and, accordingly, it's possible to design the web pages on an intranet to use the specific capabilities of the browser in question.

## What the Browser Does

The core purpose of a web browser is to connect to web servers, request documents, and then properly format and display those documents. Web browsers can also display files on your local computer, download files that are not meant to be displayed, and in some cases even allow you to send and retrieve email. What the browser is best at, however, is retrieving and displaying web documents. Each web page is written in a language called the *Hypertext Markup Language* (HTML) that includes the text of the page, a description of its structure, and links to other documents, images, or other media. The browser takes the information it gets from the web server and formats it for your display. Different browsers might format and display the same file in diverse ways, depending on the capabilities of that system and how the browser is configured.

## An Overview of Some Popular Browsers

There's a good chance you use only one browser, or two, if you use a browser on a computer and one on a mobile device. However, your website will probably be visited by a variety of browsers, and to publish on the Web successfully, you'll need to be aware of them. This section describes some of the most popular browsers on the Web. They're in no way the only browsers available, and if the browser you're using isn't listed here, don't feel that you have to use one of these. Whichever browser you have is fine as long as it works for you.

### Google Chrome

Google Chrome is currently the most popular web browser. Its market share has shown incredible growth because the browser offers great performance and stability and is updated often, plus it is used on both desktop and mobile devices. It uses the same HTML engine as Apple's Safari browser, an open source engine called WebKit. It's available as a free download at <http://www.google.com/chrome/>. You'll see Google Chrome used for the screenshots in this book, and I'll be talking about its special features for people creating websites, starting in Lesson 2.

### Microsoft Internet Explorer

Microsoft's browser, Microsoft Internet Explorer, is included with Microsoft Windows and is still the second most popular web browser. It has lost market share to other browsers because new versions are not released as often as Google Chrome and Mozilla Firefox. However, a huge number of people still use Internet Explorer, and it is the most unlike other browsers like Chrome, Firefox, and Safari. According to the website [CanIUse.com](#), Internet Explorer 10 offers 49% support of HTML5 features, and Internet Explorer 11 offers 58% support.

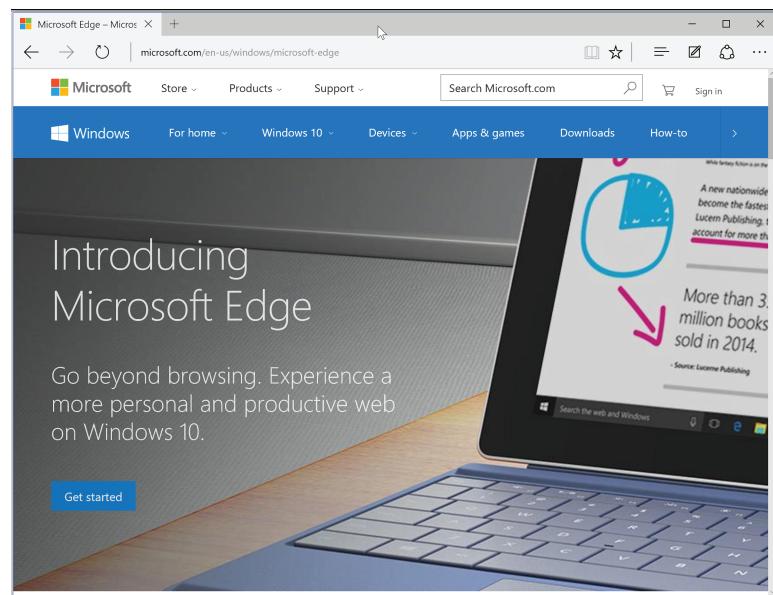
#### NOTE

If you're serious about web design, you should install all the popular browsers on your system and use them to view your pages after you've published them. That way, you can make sure that everything is working properly. Even if you don't use a particular browser on a day-to-day basis, your site will be visited by people who do. If you are interested in checking cross-browser compatibility issues, start with Microsoft Internet Explorer and Mozilla Firefox, and include Google Chrome, too.

Figure 1.5 shows Microsoft Edge—the successor to Internet Explorer—running under Windows 10.

**FIGURE 1.5**

Microsoft Edge  
(Windows 10).



One other important point to make about Internet Explorer is that the different versions differ greatly. Version 10 of Internet Explorer was released in 2012, but many users haven't upgraded from version 9, version 8, or even version 7. Internet Explorer differs widely between versions, so to get a site to work properly you need to test in each version. Web publishers have dropped support for version 6, and most have also dropped version 7, and Microsoft recommends that all users upgrade to a newer version. And in 2015 Microsoft released a new browser Microsoft Edge—the default browser for Windows 10.

## Mozilla Firefox

Mozilla Firefox is a free, open source web browser that makes up roughly 15% of the browser market as of July 2015. Netscape Navigator was the first popular commercial web browser. Version 1.0 was released in 1994. In 1998, Netscape Communications opened the source code to their web browser and assigned some staff members to work on making it better. Seven years and many releases later, the result of that effort was Mozilla Firefox. Netscape Communications, since acquired by America Online, no longer has any official ties to the Mozilla Foundation, which is now an independent nonprofit organization.

Firefox became popular in large part because it was free from the security issues that plagued Internet Explorer. In addition, a large number of Firefox extensions improve the browser experience, and Firefox has done a good job of keeping up with web standards as they have evolved. Firefox is available for Windows, Mac OS X, and Linux and is a free download at <http://www.mozilla.com/>.

1

## Apple Safari

Safari is the default browser for OS X. There is also a mobile version of this browser installed on the Apple iPhone and iPad. It is based on open source technology, and its support for web standards is at a similar level to Firefox. Right now, Safari has around 9% of the browser market share.

## Mobile Browsers

No discussion of browsers would be complete without talking about mobile browsers. The big three are Chrome, Safari, and Android. As of July 2015, Chrome has 31% of the market share for mobile browsers, with Safari and Android at 24% and 15%, respectively.

Google Chrome came on the market for mobile devices before 2014 and became the most popular browser on mobile devices in early 2015. It uses the same engine as the desktop version of the browser and offers the performance and reliability people have come to expect of Chrome. Safari is the browser Apple includes with iOS devices like the iPhone and iPad. It offers very strong HTML5 support and, apart from screen size, provides an experience very similar to a desktop browser. Similarly, Android provides a browser that also provides a high-quality web experience. All three of them are based on the WebKit rendering engine, just like Safari and Chrome for the desktop. I'll discuss the considerations that go into building sites that are friendly to mobile devices in Lesson 21, "Designing for the Mobile Web."

## Other Browsers

As of July 2015, Google Chrome has the lion's share of the market for web browsers on both desktop and mobile devices. The remaining browsers all share a relatively small slice of the pie—13% or less. For example, Opera (<http://www.operasoftware.com/>) has a niche market with only 5% share. It's small, fast, free, and available for a number of platforms, including Windows, Mac OS X, and Linux. It's also standards compliant. For UNIX users who use KDE, there's Konqueror. There are various Mozilla offshoots, such as Camino for Mac OS X. Likewise, command-line browsers such as Lynx and Links are available to provide an all-text view of web pages. There are also a number of browsers that provide access to the Web for people with various special needs. It makes sense to code to common standards to accommodate all these types of browsers.

## Web Servers

To view and browse pages on the Web, all you need is a web browser. To publish pages on the Web, you need a web server.

A *web server* is the program that runs on a computer and is responsible for replying to web browser requests for whatever content is associated with a particular URL. You need a web server to publish documents on the Web. One point of confusion is that the computer on which a server program runs is also referred to as a server. So, when someone uses the term *web server*, she could be referring to a program used to respond to requests for web pages or the computer on which that program runs.

When you use a browser to request a page on a website, that browser makes a web connection to a server using HTTP. The server accepts the connection, sends the contents of the requested files, and then closes the connection. The browser then formats the information it got from the server.

On the server side, many different browsers can connect to the same server to get the same information. The web server is responsible for handling all these requests.

Web servers do more than just serve files. They're also responsible for managing form input and for linking forms and browsers with programs such as databases running on the server.

As with browsers, many different servers are available for many different platforms, each with many different features. For now, all you need to know is what the server is there for; you'll learn more about web servers in Lesson 23, "How to Publish Your Site."

## Uniform Resource Locators

As you learned earlier, a URL is a pointer to some bit of data on the Web, be it a web document, an image, a style sheet, or a JavaScript script. You'll learn about all of these later. The URL provides a universal, consistent method for finding and accessing information.

In addition to typing URLs directly into your browser to go to a particular page, you also use URLs when you create a hypertext link within a document to another document. So, any way you look at it, URLs are important to how you and your browser get around on the Web.

URLs contain information about the following:

- How to get to the information (which protocol to use: FTP, HTTP, or file)
- The Internet hostname of the computer where the content is stored  
(www.ncsa.uiuc.edu, ftp.apple.com, netcom16.netcom.com, and so on)
- The directory or other location on that site where the content is located

1

You also can use special URLs for tasks such as sending mail to people (called *Mailto URLs*) and running JavaScript code. You'll learn all about URLs and what each part means in Lesson 6, "Working with Links."

## Defining Web Publishing Broadly

When the Web was invented, web publishing meant one thing: creating web pages as individual files and uploading them to a server so that people could view them in their browsers. Since then, pretty much everything has changed.

A few websites still include hand-coded web pages that the creator uploads, but most websites are created using software that runs on the server. Web pages have gotten more complex, as have websites. These days, most content on web pages is written using applications that live on the Web as well. For example, you can create a blog at WordPress.com and immediately begin posting content through the WordPress web interface.

Whether you're posting status updates on Twitter, writing comments on a news site, publishing a blog through a tool, or editing articles on Wikipedia, you're publishing on the Web. In most cases, you are not required to directly write HTML on your own. Generally, the pages live in templates that someone else created, and often you can format the content you create using a graphical editor or with simplified markup that enables you to avoid the use of HTML.

 Ultimately the content, however you enter it, will be converted to HTML before it is displayed to users. So if you publish something and it doesn't look right, you'll need to know HTML if you want to fix it. You'll need to be able to differentiate between the parts of the page you control and the parts that are built in to the publishing application that you're using. And if you want to take greater control of the appearance of your site, you will probably need to know HTML to update the templates that are used to give your pages their own look and feel.

So no matter what approach you take to web publishing, you will likely benefit by starting with the basics and learning how web publishing works from end to end. You may never write individual web pages by hand, but understanding how to do so will prepare you to build websites using whichever tool you ultimately choose.

## Summary

To publish on the Web, you have to understand the basic concepts that make up the parts of the Web. In this lesson, you learned three major concepts. First, you learned about a few of the more useful features of the Web for publishing information. Second, you learned about web browsers and servers and how they interact to deliver web pages. Third, you learned about what a URL is and why it's important to web browsing and publishing.

## Workshop

Each lesson in this book contains a workshop to help you review the topics you learned. The first section of this workshop lists some common questions about the Web. Next, you'll answer some questions that I'll ask you about the Web. The answers to the quiz appear in the next section. At the end of each lesson, you'll find some exercises that will help you retain the information you learned about the Web.

## Q&A

**Q Who runs the Web? Who controls all these protocols? Who's in charge of all this?**

**A** No single entity owns or controls the World Wide Web. Given the enormous number of independent sites that supply information to the Web, for any single organization to set rules or guidelines would be impossible. Two groups of organizations, however, have a great influence over the look and feel and direction of the Web itself.

The first is the *World Wide Web Consortium* (W3C), based at Massachusetts Institute of Technology in the United States and INRIA in Europe. The W3C is made up of individuals and organizations interested in supporting and defining the languages and protocols that make up the Web (HTTP, HTML, XHTML, and so on). It also provides products (browsers, servers, and so on) that are freely available to anyone who wants to use them. The W3 Consortium is the closest anyone gets to setting the standards for and enforcing rules about the World Wide Web. You can visit the Consortium's home page at <http://www.w3.org/>.

The second group of organizations that influences the Web is the browser developers themselves, most notably Google, Apple, Microsoft, and the Mozilla Foundation. The competition to be the most popular and technically advanced browser on the Web can be fierce. A group of people and companies interested in the future of the Web have created an organization called the Web Hypertext Application Technology Working Group (or WHATWG). The WHATWG, along with the W3C, wrote the HTML5 specification.

Going forward, the WHATWG has abandoned version numbers for the HTML specification entirely. Instead, HTML will be a “living standard” and incorporate both experimental and widely supported features. The goal is to make sure that the specification evolves to match the features that browser makers have agreed to add to their browsers. If a proposed feature does not reach consensus, it is removed from the specification. This is an attempt to prevent the problems of the past where the process of creating the HTML specification diverged from the work the browser makers were doing.

**Q I've heard that the Web changes so fast that it's almost impossible to stay current. Is this book doomed to be out-of-date the day it's published?**

**A** Although it's true that things do change on the Web, the vast majority of the information in this book will serve you well far into the future. HTML is as stable now as it has ever been, and once you learn the core technologies of *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS), and JavaScript, you can add on other things at your leisure.

## Quiz

1. What's a URL?
2. What's required to publish documents on the Web?

## Quiz Answers

1. A URL, or uniform resource locator, is an address that points to a specific document or bit of information on the Internet.
2. You need access to a web server. Web servers, which are programs that serve up documents over the Web, reply to web browser requests for files and send the requested pages to many different types of browsers. They also manage form input and handle database integration.

## Exercises

1. Start thinking more about web publishing as you surf the Web. Look at how URLs are constructed. Pay attention to how the pages are constructed. Soon you'll understand how these pages are built from the inside out.
2. Download a different browser than the one you ordinarily use and try it out for a while. If you're using Internet Explorer, try out Firefox, Chrome, Safari, or even a command-line browser such as Lynx or Links. To really see how things have changed and how some users who don't upgrade their browser experience the Web, download an old browser from <http://browsers.evolt.org/> and try it out.

*This page intentionally left blank*

## LESSON 2

# Getting Your Tools in Order

When you start on a project, whether it's writing a book or painting a room, you don't just jump in and grab a brush or start typing. You come up with a plan to complete the project and gather the materials you need to get the work done. Your plan may be as simple as deciding to paint the walls before you paint the ceiling, or it may be as complex as a detailed outline of everything you plan to write in your book.

The same goes for materials. If you're painting a room, you need brushes, paint, and maybe something to prevent getting paint where you don't want it. Likewise, if you're writing a book, you'll want to install a word processor and perhaps gather the research materials you need to support your writing. Just as with most other projects, the process of writing and designing web pages takes some planning and thought before you start flinging text and graphics around and linking them wildly to each other. Likewise, you'll want to make sure you have everything you need on your computer to build web pages, as well as a place on the Web to host your website when you're finished.

To prepare to publish on the Web, you must

- Learn the differences between a web server, a website, a web page, and a home page.
- Set up your computer so that you can start creating web pages.

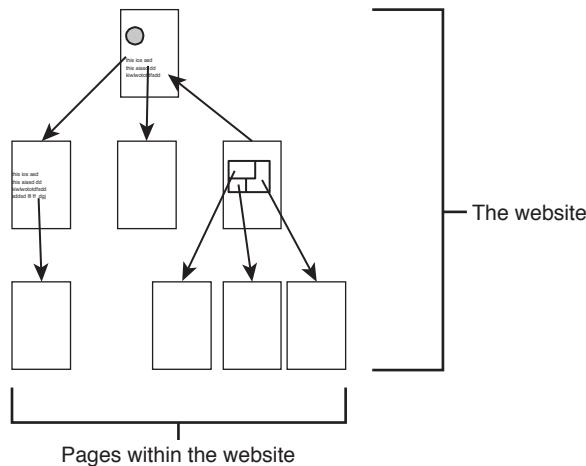
## Anatomy of a Website

First, here's a look at some simple terminology I use throughout this book. You need to know what the following terms mean and how they apply to the body of work you're developing for the Web:

- **Website**—A collection of one or more web pages linked together in a meaningful way that, as a whole, describes a body of information or creates an overall effect (see Figure 2.1).

**FIGURE 2.1**

Websites and pages.



- **Web server**—A computer on the Internet or an intranet that delivers web pages and other files in response to browser requests. (An intranet is a network that uses Internet protocols but is not publicly accessible.)
- **Web page**—A single document on a website, usually consisting of a *Hypertext Markup Language* (HTML) document and any items that are displayed within that document, such as inline images or style sheets.
- **Home page**—The entry page for a website, which can link to additional pages on the same website or pages on other sites.

Each website is hosted on a web server. Throughout the first few lessons in this book, you'll learn how to develop well thought-out and well-designed websites. Later, you'll learn how to publish your site on an actual web server.

A *web page* is an individual element of a website in the same way that a page is a single element of a book or a newspaper (although, unlike paper pages, web pages can be of any

length). Web pages sometimes are called *web documents*. Both terms refer to the same thing. A web page consists of an HTML document and all the other components that are included on the page, such as images or other media.

**NOTE**

Most websites aren't built out of individual pages these days. Rather, they are created using applications that publish web content stored in a database of some kind through a common set of templates. The URLs on the site act as input for the publishing application. In this book, you'll still be creating web pages in the traditional sense, because it's the easiest way to learn.

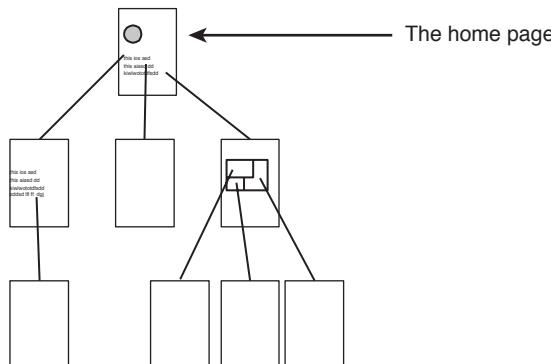
If you're publishing a website, the home page is the first or topmost page on your website. It's the intended entry point that provides access to the rest of the pages on the site (see Figure 2.2).

**CAUTION**

Most of your customers will access your site through your home page, but some will enter your site through other pages. The nature of the Web is that people can link to any page on your site. If you have interesting information on a page other than your home page, people will link directly to that page. On the other pages of your site, you shouldn't assume that the visitor has seen your home page.

**FIGURE 2.2**

A home page.



A home page often contains an overview of the content of the website, available from that starting point—for example, in the form of a table of contents or a set of icons. If

you don't have too much content, you might include everything on that single page—making your home page and your website the same thing. A personal home page might include a link to a person's resumé and a link to his Twitter account and his photos on Flickr. A restaurant's home page will likely include the restaurant's hours and location, and links to the menu and directions to the restaurant. A corporate home page usually describes what the company does and contains links like "About the Company," "Products and Services," and "Customer Support."

## Setting Up Your Computer for Web Publishing

First of all, if you just want to post some words or pictures on a web page, you don't really need to do anything to set up your computer. You can just open a browser, find a site that enables you to publish your content like Tumblr or Wix, and then publish from within the browser.

If you want to learn how to create websites from scratch, set up your own computer so that you can create web pages and view them locally. The only two tools you absolutely must have to experiment with web publishing are a text editor and a web browser. You learned a bit about web browsers in the previous lesson, so let's talk about text editors first.

### Text Editors

HTML files are plain-text files and should be edited using a tool that works with plain-text files. What this essentially means is that you are going to learn HTML, you shouldn't be editing your files with a word processing application like Microsoft Word or an online application like Google Docs. Those types of programs allow you to edit files in what will be their final format and then save the results in a document format like Microsoft Word's proprietary format. Confusingly, these types of applications will also enable you to save your documents as HTML documents. Doing so may be sufficient to meet your needs in terms of producing a document but won't teach you the first thing about HTML, which is your goal.

If you've used text editors before, chances are you already have a favorite. People tend to be highly opinionated about what makes a good text editor. If you haven't used one at all or haven't used one much, you'll need a recommendation. To get started, you can use the text editor that's provided with your computer's operating system; they all have one.

If you're a Windows user, you can use the Notepad application. If you are using OS X, you can start with TextEdit. If you are a Linux user, you can start with vi or Emacs.

NotePad, TextEdit, and vi offer very limited functionality, and if you do a lot of text editing, you'll want to track down another more powerful application to do your text editing. Here's a list of a few editors often used by people who create websites:

- Komodo Edit is a free, open source version of the popular Komodo IDE. It runs on Windows, Macintosh, and Linux and offers a lot of features of an IDE. You can download it at <http://komodoide.com/komodo-edit/>.
- HTML-Kit is a popular text editor specifically for web pages for Windows. You can download it at <http://www.htmlkit.com/>. You can use an older version for free or you can pay for the latest and greatest.
- Notepad++ is a free, open source text editor for Windows that is very popular. You can download it at <http://notepad-plus-plus.org/>.
- TextWrangler is a popular, free text editor for OS X. It was created by Bare Bones Software, and you can download it at their website at <http://www.barebones.com/products/textwrangler/>. You may also be interested in BBEdit, a more powerful text editor with a licensing fee.
- Coda is a text editor specifically for people creating web pages by Panic. It includes a lot of development features like source control and database connectivity. You can find it at <http://panic.com/coda/>. It also has a licensing fee.

You'll want to find your text editor and open the application. If you're using TextEdit on OS X, make sure that it's in plain-text mode. If the document window has controls that let you choose a font or apply other formatting, go to the Format menu and select Make Plain Text. Once you have your editor open, you can type in some stuff and, if you like, save the file you're editing. The main thing to make note of is that simply typing in characters with your keyboard is the only thing you're able to do. You have no formatting options whatsoever—that's what's meant by *plain text*.

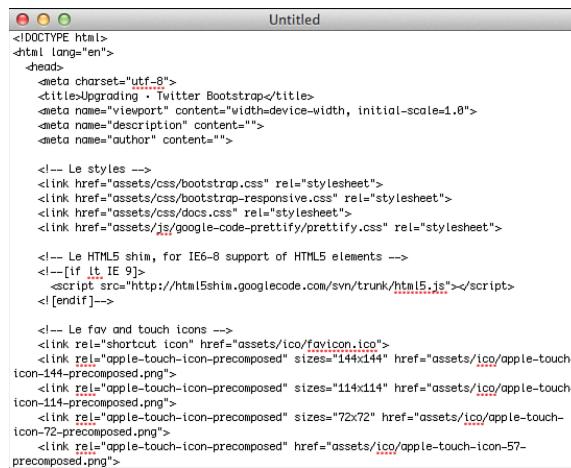
#### **NOTE**

You can work through every lesson in this book using NotePad or TextEdit, but most web developers find that using a more powerful tool improves their productivity significantly. Many provide highlighting that makes your documents easier to read. All of them also enable you to have multiple documents open at once and enable you to treat a group of files as a project. It would be tough to find a new tool that suits you before you have even started, but I would encourage you to look into different editors as you make your way through the book. And many of the commercial editors have free trials, so you can try them out before you buy.

Figure 2.3 is a screenshot of the OS X text editor TextEdit. It is notable mainly due to the fact that it has no text formatting menu or toolbar at all. This is what you're looking for in a text editor; it should enable you to edit the contents of the file without applying formatting of any kind.

**FIGURE 2.3**

TextEdit on OS X.



The screenshot shows the OS X TextEdit application window. The title bar reads "Untitled". The main content area displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Upgrading · Twitter Bootstrap</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">

    <!-- Le styles -->
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/bootstrap-responsive.css" rel="stylesheet">
    <link href="assets/css/docs.css" rel="stylesheet">
    <link href="assets/js/google-code-prettify/prettify.css" rel="stylesheet">

    <!-- Le HTML5 shim, for IE6-8 support of HTML5 elements -->
    <!--[if lt IE 9]>
      <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
    <![endif]-->

    <!-- Le fav and touch icons -->
    <link rel="shortcut icon" href="assets/ico/favicon.ico">
    <link rel="apple-touch-icon-precomposed" sizes="144x144" href="assets/ico/apple-touch-icon-144-precomposed.png">
    <link rel="apple-touch-icon-precomposed" sizes="114x114" href="assets/ico/apple-touch-icon-114-precomposed.png">
    <link rel="apple-touch-icon-precomposed" sizes="72x72" href="assets/ico/apple-touch-icon-72-precomposed.png">
    <link rel="apple-touch-icon-precomposed" href="assets/ico/apple-touch-icon-57-precomposed.png">
```

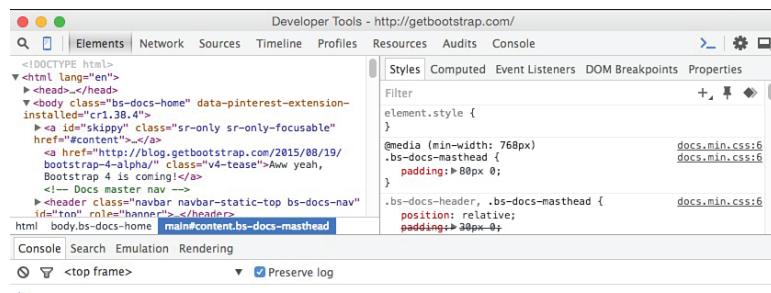
## A Web Browser

As mentioned in the previous lesson, a number of popular web browsers are available, and you can use any that you like to surf the Web on a day-to-day basis. However, as you're working through the lessons in this book, I'm going to recommend one browser in particular: Google Chrome. The main reason is that Google Chrome offers a number of powerful tools aimed at helping people create websites. Other browsers have similar tools, but I'm going to make reference to the Google Chrome Developer Tools specifically in the text, and you'll find it easier to follow along if you're using Google Chrome as well. If you feel confident, you can choose another browser if you prefer. (You'll need to translate the parts where I mention Google Chrome to your own browser, but I'd encourage you to download Google Chrome and work through the next section regardless of which browser you plan to ultimately use, especially if you're completely unfamiliar with these kinds of tools). You can download Google Chrome at <http://google.com/chrome>.

# Using the Google Chrome Developer Tools

After you've downloaded and installed Google Chrome, open the application, and navigate to <http://getbootstrap.com/>. Bootstrap is a generic framework for web pages and is discussed later. For now, it's useful because the source code for the web pages was written to be easily readable. In Chrome's View menu, open the Developer submenu, and then click Developer Tools. At this point, the Developer Tools will open, as shown in Figure 2.4.

**FIGURE 2.4**  
The Google Chrome Developer Tools.



## TIP

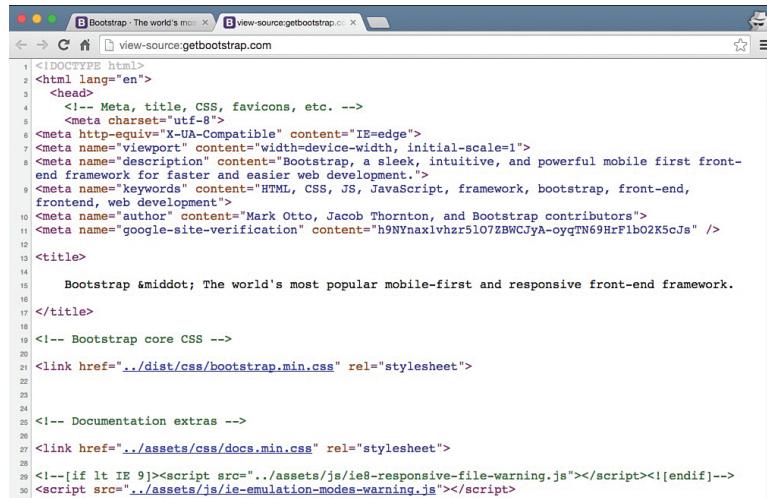
There is also a keyboard shortcut to open the Developer Tools. On Windows, you can open them by pressing Control+Shift+I. On OS X, you can open them with Command+Option+I. You'll find yourself using the Developer Tools a lot, so it's definitely worth memorizing the keyboard shortcut.

The Developer Tools opens as a panel in the browser, covering the bottom of the web page. If you prefer, you can click the button on the upper right to detach the Developer Tools from the browser window. This allows you to see more in both windows, but you'll have to switch between them. You can also move the tools to the side of your browser window rather than the bottom if you prefer. Position the Developer Tools however you feel the most comfortable.

From the earliest days of the Web, browsers have supported a feature called “View Source” that displays the actual HTML source code for the web page that you’re viewing. In Google Chrome, you can view the source for the current page by selecting View Source from the Developer submenu of the browser’s View menu. The source for <http://getbootstrap.com/> appears in Figure 2.5.

**FIGURE 2.5**

The source code for <http://getbootstrap.com/>.



```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <!-- Meta, title, CSS, favicons, etc. -->
5     <meta charset="utf-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1">
8     <meta name="description" content="Bootstrap, a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development.">
9     <meta name="keywords" content="HTML, CSS, JS, JavaScript, framework, bootstrap, front-end, frontend, web development">
10    <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
11    <meta name="google-site-verification" content="h9Nnaxlvhzr5l07zBWCJyA-cyqTN69HrFlb02K5cJS" />
12
13    <title>
14      Bootstrap &middot; The world's most popular mobile-first and responsive front-end framework.
15    </title>
16
17    <!-- Bootstrap core CSS -->
18
19    <link href="../dist/css/bootstrap.min.css" rel="stylesheet">
20
21
22
23
24
25    <!-- Documentation extras -->
26
27    <link href="../assets/css/docs.min.css" rel="stylesheet">
28
29    <!--[if lt IE 9]><script src="../assets/js/ie9-responsive-file-warning.js"></script><![endif]-->
30    <script src="../assets/js/ie-emulation-modes-warning.js"></script>

```

The Developer Tools are a much more powerful extension of this concept. The Developer Tools have a number of tabs. When you open them, the Elements tab is displayed. This tab contains the source of the page, sort of. When a browser downloads a web page, it transforms it so that the engine that formats the HTML and presents it can understand it. Depending on the validity of the web page, this transformation is pretty minor. The Elements tab presents the HTML as the browser sees it. View Source shows the actual HTML that the browser downloaded, so if you compare the contents of the View Source window with the contents of the Elements tab, you’ll see a few differences that illustrate what I’m talking about.

Don’t worry about what any of the actual HTML does right now, I’ll dig into that soon enough. For now, just focus on the tool. When you move your mouse over the elements in the Elements tab, the part of the web page associated with the element under the mouse will be highlighted so that you can see how the HTML corresponds to the HTML source. When you click one of the elements, the panes to the right of the window are updated with the style information for that element. Later, when you start working with Cascading Style Sheets, this feature will be really helpful because it shows exactly how the browser interprets your styles.

Finally, on the bottom row, you'll find a number of buttons. The first is the button that detaches or reattaches the tools window. The next opens the JavaScript console, which is discussed starting in Lesson 17, "Introducing JavaScript." The next button looks like a magnifying glass. If you click it, you can then click content on the web page, and the HTML element corresponding to it will be selected in the Elements tab. This is useful when you want to inspect a particular element on the page.

Finally, the next buttons show the nesting order of the tags for the selected elements. These are useful for moving through the structure of the web page. You'll learn more about how pages are structured in lessons to come.

2

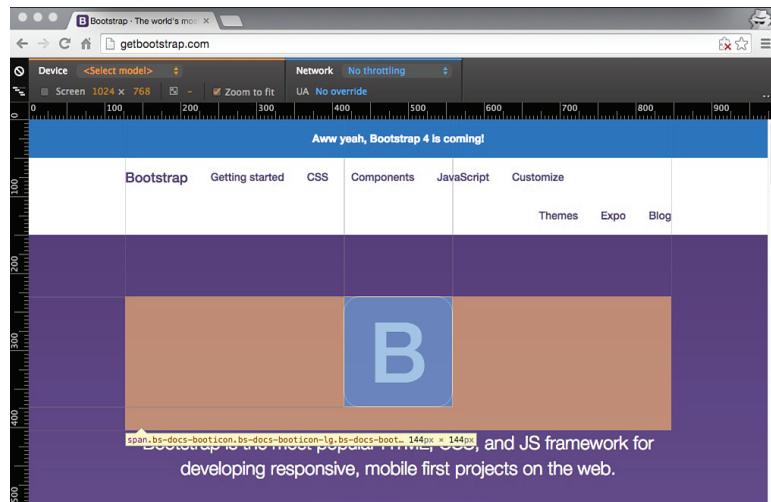
## Exercise 2.1: Using the Inspector



Before moving on, it's worth seeing exactly how the Inspector works to see how to use the Developer Tools to find specific elements on the page in the Elements view. If you're not viewing <http://getbootstrap.com/> in your browser, go ahead and open it, and then open the Chrome Developer Tools. You may want to go ahead and try the keyboard shortcut mentioned earlier in a tip. As I said, as you progress you'll find yourself doing this a lot.

Once the browser tools are open, click the "inspect" button in the Developer Tools. It's the first button on the left in the bottom row—a magnifying glass. The button will turn blue, indicating that the Developer Tools are ready to inspect the element that you click. At this point, move the mouse over the browser window until you locate the large heading near the top of the page. The browser window should look something like the one in Figure 2.6.

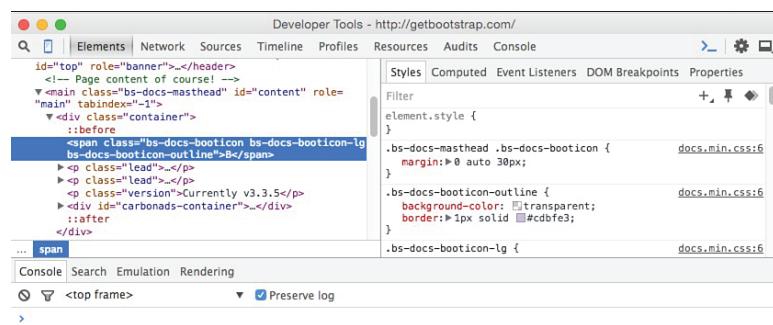
**FIGURE 2.6**  
Inspecting the Bootstrap page.



- ▼ As you can see, the heading is highlighted, and some information about the size of the element is provided in a tool tip. This is the element I want you to inspect, so you should locate it in your browser and click it. As you saw in Figure 2.4, the source of the page shown in the Elements is mostly collapsed when you initially open the Developer Tools. When you inspect an element, the source is expanded enough to show you the HTML source corresponding to whatever it was on the page that you clicked. The Developer Tools window should now be displaying the HTML tags used to create the heading you clicked. It will look like Figure 2.7.

**FIGURE 2.7**

Inspecting the B icon in the Developer Tools.



I don't expect to understand much of what you see in the Developer Tools window yet, but let me go over the highlights. In the left panel, you'll see the actual HTML source for the page. In the right column, you'll find style information that shows why the selected element looks as it does. In the bottom row, you can see where the selected element falls within the structure of the page. The selected element is the "span" around the Bootstrap B at the right end of the bottom bar. It is four levels deep in the page and has three CSS classes applied to it.

You'll find yourself falling back on the Developer Tools a lot, especially when things don't look like you'd expect them to on the page. It shows how the browser sees your page and makes it easy to drill down to exactly the element you want to find, which is especially useful as your pages grow larger and more complex.



## What Do You Want to Do on the Web?

This question might seem silly. You wouldn't have bought this book if you didn't already have some idea of what you want to publish. But maybe you don't really know what you want to put on the Web, or you have a vague idea but nothing concrete. Maybe it has suddenly become your job to work on the company website, and someone handed you this book and said, "Here, this will help." Maybe you're a software developer who's suddenly in charge of building a web interface for a product or building a web application. Maybe you just want to do something similar to some other web page you've seen and thought was particularly cool.

What you want to put on the Web is what I refer to throughout this book as your content. *Content* is a general term that can refer to text, graphics, media, forms, and so on. If you tell someone what your web pages are about, you're describing your content.

The only thing that limits what you can publish on the Web is your own imagination. In fact, if what you want to do seems especially wild or half-baked, that's an excellent reason to try it. The most interesting websites are the ones that stretch the boundaries of what the Web is supposed to be capable of.

You might also find inspiration in looking at other websites similar to the one you have in mind. If you're building a corporate site, look at the sites belonging to your competitors and see what they have to offer. If you're working on a personal site, visit sites that you admire and see whether you can find inspiration for building your own site. Decide what you like about those sites and you want to emulate and where you can improve on those sites when you build your own.

These days, the barriers to building many kinds of websites are extremely low. If you want to publish text and photos, you can use one of any number of free blogging sites to set up a site in minutes, as long as blogging software suits your needs. Experimenting is easier than ever. Try something, see whether it takes off, and then build from there.

If you really have no idea of what to put up on the Web, don't feel that you have to stop here, put this book away, and come up with something before continuing. Maybe by reading through this book, you'll get some ideas (and this book will be useful even if you don't have ideas). I've personally found that the best way to come up with ideas is to spend an afternoon browsing on the Web and exploring what other people have done.

## Wireframing Your Website

The next step in planning your website is to figure out what content goes on which pages and to come up with a scheme for navigating between those pages. If you have a lot of

content that needs to be linked together in sophisticated ways, sitting down and making a specific plan of what goes where will be incredibly useful later as you develop and link each individual page.

## What's Wireframing, and Why Do I Need It?

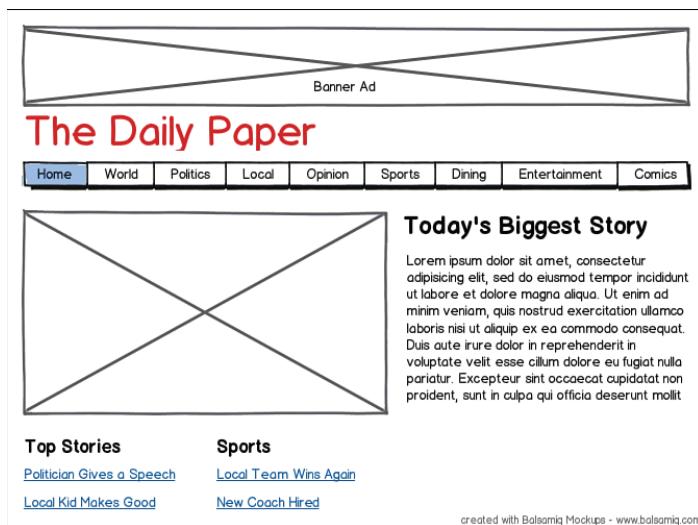
*Wireframes* provides a rough outline of what the website will look like when it's done, showing which content will appear on which pages and how they will be connected together. With that representation in hand, you can develop each page without trying to remember exactly where that page fits into the overall website and its often complex relationships to other pages.

In the case of really large sites, wireframes enable different people to develop various portions of the same website. With clear wireframes, you can minimize duplication of work and reduce the amount of contextual information each person needs to remember.

For smaller websites, or websites built using content management applications that provide a specific structure, wireframes might be unnecessary. For larger and more complex projects, however, the existence of wireframes can save enormous amounts of time and frustration. If you can't keep all the parts of your content and their relationships in your head, consider creating a wireframe.

So, what do wireframes look like? Generally speaking, they are collections of documents or images, each of which represents a certain type of page on a website. The documents contain a rough diagram of the page, illustrating how the various components of the page will be positioned, how much space they should take up, and what function they will serve. For example, the wireframes for a newspaper website would include a diagram of the home page, the home page for sections of the paper, and a wireframe for article pages. The wireframes might also include the registration form for the site and a page that can be used to purchase advertisements. An example wireframe created using a tool called Balsamiq is included in Figure 2.8.

Don't feel that your wireframes have to be pretty or built in specific wireframing software. The point of wireframing is that it organizes your web pages in a way that works for you. If you like index cards and string, work with these tools. If a simple outline on paper or on the computer works better, use that instead.



## Hints for Wireframing

Some things to think about when developing your wireframes are as follows:

- **Which topics will go on each page?**

Trying to figure out how much information to put on one page can be tricky. Some websites put all of their content on one long, cleverly designed page. Others split it up between a great many pages. Still others use modern techniques to dynamically load parts of the page on demand without ever really moving from one page to another. Without getting too fancy, your best bet is to organize your content so that each page consists of information on a single topic. If your pages become more than a few screens long, it might be time to split them into logical subtopics.

- **What are the primary forms of navigation between pages?**

What links will you need for your visitors to navigate from page to page? They are the main links in your document that enable your visitors to accomplish the goals you defined in the first section. Links for forward, back, up, down, and home all fall under the category of primary navigation.

- **What alternative forms of navigation are you going to provide?**

In addition to the simple navigation links, some websites contain extra information that's parallel to the main web content, such as a glossary of terms, an alphabetic index of concepts, copyright information, or a credits page. Consider these extra

forms of information when designing your plan, and think about how you’re going to link them into the main content.

### ■ What will you put on your home page?

Because the home page is the starting point for the rest of the information in your website, consider what sort of information you’re going to put on the home page. A blog? A general summary of what’s to come? A list of links to other topics? Whatever you put on the home page, make sure that it’s compelling enough so that members of your intended audience want to stick around.

### ■ How will visitors to inner pages establish context?

Unless your website requires customers to register to view your content, there’s a good chance that users could arrive on any page on your site by way of a search engine. It’s important to make sure that customers can figure out which site they’re on and that there’s more information that they may also be interested in. You can generally establish this context through your design and navigation.

### ■ What are your goals?

As you design the framework for your website, keep your goals in mind, and make sure that you aren’t obscuring your goals with extra information or content.

#### TIP

Several utilities and packages can assist you in creating wireframes. Some free tools include Mockingbird (<http://gomockingbird.com/>), Denim (<http://dub.washington.edu:2007/denim/>), and Gliffy (<https://www.gliffy.com/uses/wireframe-software/>). In addition, several mobile apps for iOS and Android devices help you create website mockups.

## Web Hosting

At some point, you’ll want to move the websites you create from your local computer to a server on the Internet. Before doing so, you must decide exactly what kind of hosting arrangement you want. The simplest approach is to get a web hosting account that enables you to upload your HTML files, images, style sheets, and other web content to a server that’s visible on the Web. This approach enables you to easily create web pages (and websites) locally and publish them on the server without making changes to them.

## Using a Content-Management Application

The other option is to use an application to publish content on the Web. This can make more sense if your idea for a website falls into an existing category with publishing

tools available for it. For example, if you want to publish a blog, you can use sites like TypePad (<http://typepad.com/>), Blogger (<http://blogger.com>), WordPress (<http://wordpress.com/>), or Tumblr (<http://tumblr.com>), among many others. The advantage of these applications is that it's easy to set up a site, pick a theme, and start publishing content on the Web through a web interface. There's no need to build the web pages by hand, set up a hosting account, or even deal with editing files by hand.

There are also online tools like Wix (<http://www.wix.com>), Squarespace (<http://squarespace.com>), and Weebly (<http://www.weebly.com>) that let you build a more open format website than a blog. These applications make it easy to create a website using their many templates, and they include other features like domain names, ecommerce, image and multimedia collections, and more. The advantage of using these applications is that they are easy to set up and create more professional-looking sites than blogging platforms.

Generally with either of these types of applications, all you need to do to get started is fill out a form, choose a URL, and pick a theme for your website. Then you can enter your content by way of forms, enabling you to avoid writing the HTML for the pages yourself. Some of them even include WYSIWYG editors so that you can format the content you enter without using HTML.

However, that doesn't mean that you don't need to learn anything about HTML or *Cascading Style Sheets* (CSS). Even if you're not creating the pages by hand, you'll still need to understand how pages are structured when you start entering content or modifying themes yourself. If you don't understand how web pages are built, you won't know how to track down and fix problems with the markup on your website, whether you're responsible for writing it or not.

For most people taking their first steps into web publishing, using an application to get started is the best approach, because it enables you to start putting the content you're interested in on the Web immediately without figuring out too many things for yourself. However, people run into limitations in these applications that leave them wanting to take more control of their websites and go further on their own. This book will help you do so.

## **Setting Up Your Own Web Hosting**

If you do want to create and upload your own web pages, you'll need to choose a company that will provide you with the space you need. There are a huge number of hosting companies that provide web space to people who want to launch their own websites. Companies like DreamHost (<http://dreamhost.com>) and Pair.com (<http://pair.com>) have been in the hosting business for many years and offer a variety of affordable hosting

plans, but there are plenty of other options, too. Many people subscribe to hosting plans from the company that they use to register the domain name for their website or go with hosting companies that are in their local area.

If you choose to go this route, the steps for going from setting up a hosting account to making your pages available on the Web are as follows:

1. Optionally, register a domain name. If you want your website to appear at a URL like mycoolsite.com or mycompany.com, you'll need to register that domain name if you haven't already. There are a number of domain registrars; just enter "domain registration" in your favorite search engine to see a large number of ads and search results for companies that offer domain registration.
2. Pick out a web hosting company and sign up for an account. If you're going to be putting your pages on an internal or external server belonging to your employer or your school, you won't need your own hosting. But if you're creating a new website that will be available on the Internet, you'll need some sort of hosting arrangement.
3. Associate your domain name with your new website, if you have registered one. Your domain registrar and hosting company should provide instructions for setting it up so that your domain name points to your hosting account. That way when users enter your domain name in a URL, they'll get the content that you upload to your server.
4. Start uploading your content. Once your web hosting is set up, you can use whatever tool you prefer to start uploading web content to the server. Many hosts provide a web interface that will allow you to upload content, but most hosts will also let you use a file transfer tool that supports *File Transfer Protocol* (FTP), *Secure Copy* (SCP), or *Secure FTP* (SFTP) to get your files to the server.

There will be a much more extensive discussion of web hosting and how to publish your site in Lesson 23, "How to Publish Your Site," but I wanted to give you a head start if you're eager to start publishing on the Web.

## Summary

In this lesson, I explained how to get set up to productively work on web pages. You learned about how to use the Developer Tools built in to Google Chrome to assist in working on web pages and how to find a text editor that you can use to create web pages. You also learned about setting goals for your website and about finding hosting for your site. I also explained how wireframes are used to create a map of your website before you start creating it in HTML.

## Workshop

The first section of the workshop lists some of the common questions people ask while planning a website, along with an answer to each. Following that, you have an opportunity to answer some quiz questions yourself. If you have problems answering any of the questions in the quiz, go to the next section, where you'll find the answers. The exercises help you formulate some ideas for your own website.

### Q&A

- Q Getting organized seems like an awful lot of work. All I want to do is make something simple, and you're telling me I have to have plans and wireframes. Are all the steps listed here really necessary?**
- A** If you're doing something simple, you won't need to do much, if any, of the stuff I recommended in this lesson. However, if you're talking about developing two or three interlinked pages or more, having a plan before you start will really help. If you just dive in, you might discover that keeping everything straight in your head is too difficult. And the result might not be what you expected, making it hard for people to get the information they need out of your website as well as making it difficult for you to reorganize it so that it makes sense. Having a plan before you start can't hurt, and it might save you time in the long run.
- Q You talked a lot in this lesson about organizing topics and pages, but you said nothing about the design and layout of individual pages. Why?**
- A** I discuss design and layout later in this book, after you've learned more about the sorts of layout that HTML (the language used for web pages) can do and the stuff that it just can't do.
- Q What if I don't like any of the basic structures you talked about in this lesson?**
- A** Then design your own. As long as your visitors can find what they want or do what you want them to do, no rules say you *must* use a hierarchy or a linear structure. I presented these structures only as potential ideas for organizing your web pages.

### Quiz

1. How would you briefly define the meaning of the terms *website*, *web server*, and *web pages*?
2. In terms of web publishing, what's the meaning of the term *home page*?
3. Regardless of the navigation structure you use in your website, there's one link that should typically appear on each of your web pages. What is it?
4. What's the purpose of a wireframe?

## Quiz Answers

1. A *website* is one or more web pages linked together in a meaningful way. A *web server* is the actual computer that stores the website (or, confusingly enough, the piece of software that responds to requests for pages from the browser). *Web pages* are the individual elements of the website, like a page is to a book.
2. A *home page*, in terms of web publishing, is the entry point to the rest of the pages in your website (the first or topmost page).
3. You should try to include a link to your home page on each of the pages in your website. That way, users can always find their way back home if they get lost.
4. A wireframe provides an overall outline of what the website will look like when it's done. It helps organize your web pages in a way that works for you. It is most beneficial for a larger website.

## Exercises

1. Come up with a list of several goals that your visitors might have for your web pages. The clearer your goals are, the better.
2. After you set your goals, visit sites on the Web that cover topics similar to those you want to cover in your own website. As you examine the sites, ask yourself whether they're easy to navigate and have good content. Then make a list of what you like about the sites. How would you make your website better?