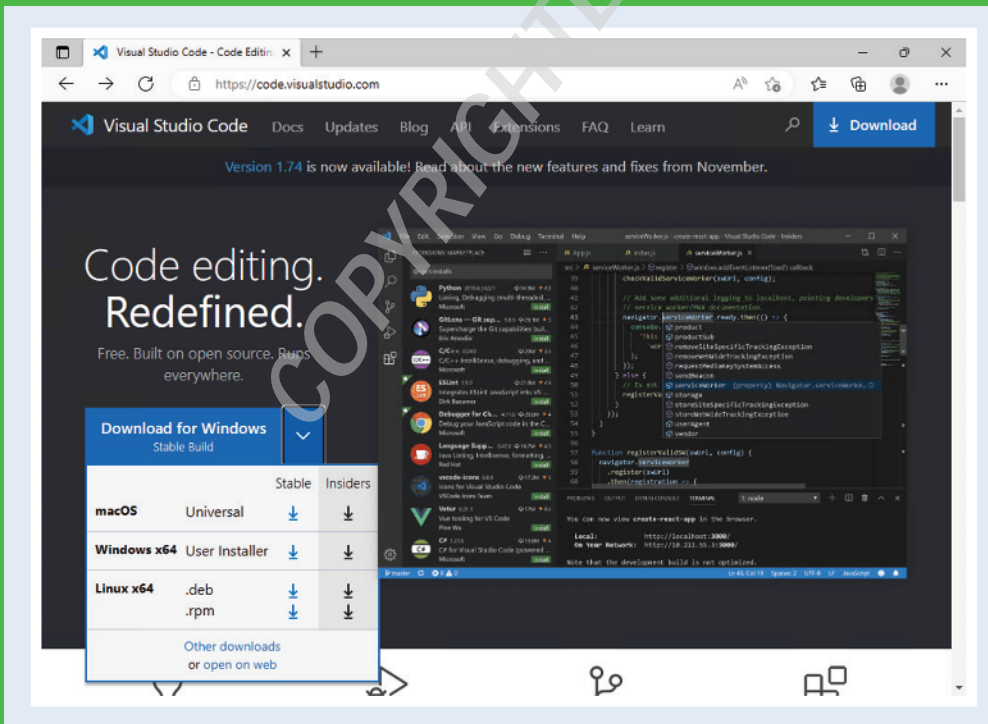


CHAPTER 1

Getting Ready to Create Websites

In this chapter, you get ready to create your website or websites. You learn the essentials of how the Web works and the technologies that power it, plan your website, and install the apps you will need to create the website. You also get a domain name and web hosting if you do not already have them.



Grasp How the Web Works	4
Understanding HTML, CSS, and Responsive Web Design . . .	6
Understanding Static and Dynamic Web Pages	8
What Is a Responsive Website?	9
Understanding Tools for Creating Web Pages	10
Prepare to Create Your Website	12
Install Visual Studio Code	14
Meet and Configure Visual Studio Code	16
Install GIMP	20
Install the Major Browsers	22
Create a Folder Structure for Your Website	24

Grasp How the Web Works

The World Wide Web, nowadays usually just called the Web, consists of a vast number of websites accessible through the Internet using a web browser. Each website contains one or more web pages — usually, many more. Each web page is identified by a unique address called a *Uniform Resource Locator*, or URL. To request a web page, the user enters its URL in a web browser app, either by typing the URL or by clicking a link. The web server hosting the website transmits the requested page to the browser, which displays the contents for the user to view.

What Is a Web Page?

A *web page* is a digital document that is accessed through the Web using a web browser app. Web pages are components of websites, discussed next, which are hosted on web servers, discussed later.

Web pages can contain text, images, audio or video files, and other digital resources, such as documents that visitors can download. Web pages are arranged and formatted using Hypertext Markup Language, HTML or short, and Cascading Style Sheets, CSS. Web pages contain contents that can be static or dynamic; they may also contain interactive features, such as forms, that enable visitors to input data or interact with the content.

What Is a Website?

A *website* is a collection of web pages hosted on a web server and made accessible to web browsers via the Web. A website typically contains multiple pages that are connected to each other by hyperlinks, forming a coherent structure that lets visitors navigate quickly between the various areas of the website.

A website typically aims to serve a specific purpose. For example, a personal website might showcase the owner's interests and talents; an organization's website might explain that organization's purpose and aims and encourage visitors to join; or a company's website might present the company in the best possible light and provide ways to buy its products.

What Is a Web Server?

A *web server* is software that responds to requests from web clients, such as web browsers, and returns content if it is available and permitted. A web server stores web pages, images, videos, and other content so that it can serve them to clients.

A web server can run on almost any computer hardware, from diminutive computers such as the Raspberry Pi series up to dedicated server machines deployed in full-scale facilities called *server farms*. As of this writing, many web servers are deployed on cloud-based infrastructure, such as Amazon Web Services, AWS, or Microsoft Azure.

A web server can run on just about any computer operating system, including Windows, macOS, Linux, and the mobile operating systems iOS, iPadOS, and Android.

Web servers are a critical part of Internet infrastructure and deliver web content to users throughout the world.

What Is a Web Browser?

A *web browser* is an app used to access and display web pages and other content on the Web. Using a web browser, you can go to a web page either by typing or pasting its address or by clicking a link. Web browsers use Hypertext Transport Protocol, HTTP, or its secure variant, Hypertext Transport Protocol Secure, HTTPS, to request web pages from web servers and then display the content in the browser window.

Popular web browsers include Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple's Safari. These browsers have many features to make browsing easier and faster, such as bookmarks and tabbed browsing.

How Does a Web Browser Find the Web Server Hosting a Website?

When you enter a website's URL into the browser's address box, the browser uses the Domain Name System, DNS, to discover the Internet Protocol address, or IP address, for the web server hosting the website. DNS uses a hierarchical system of servers to organize, store, and return the IP address associated with each domain name.

A *domain name* is a text-based identifier that represents a unique location on the Internet. For example, `www.wiley.com` is the domain name of the website for John Wiley & Sons, Inc., publisher of this book and many others. A domain name consists of multiple parts. The rightmost part is the top-level domain, or TLD — in this case, `.com`. Moving toward the left, the next part — in this case, `wiley` — is the second-level domain. The next part, `www`, is the subdomain.

Understanding HTML, CSS, and Responsive Web Design

Before you start creating pages for your website, you will likely find it helpful to understand the essentials of HTML and CSS, the two languages with which you will be working throughout this book. This section introduces you to HTML and CSS. It also gives you an executive overview of responsive web design, an approach intended to make websites equally accessible by different types of devices with different screen sizes and resolutions.

HTML standards and CSS standards are developed and maintained by the World Wide Web Consortium, W3C, with contributions from many companies and organizations, including the makers of the major browsers.

What Is HTML?

HTML is the abbreviation for Hypertext Markup Language, a language used to create web pages. *Hypertext* means text that includes hyperlinks to other locations on the same page or to other pages, so when you click the linked text, the browser displays the linked location or page.

HTML enables you to “mark up” text and other elements with codes that specify how the elements appear. For example, you can mark up a paragraph as a first-level heading by enclosing it in the appropriate HTML codes, which are `<h1>` at the beginning and `</h1>` at the end:

```
<h1>This Is a Heading 1 Paragraph</h1>
```

Similarly, you can mark up a paragraph as being regular “paragraph” text by enclosing it in `<p>` and `</p>` codes:

```
<p>This is a paragraph of regular text.</p>
```

The nearby illustration shows how this heading and paragraph look using the Google Chrome browser’s default styles for the `h1` element (A) and the `p` element (B). You can control the formatting by defining and applying styles of your own.

HTML is currently at version 5, which is generally referred to as HTML5. But rather than being a fixed version, HTML5 is what is called a *living standard*, with development continuing and new features being released. So although HTML5 was first released in January 2008 and went through a major update in October 2014, it is still the current version as of this writing in April 2023 — and it looks likely to remain the current version for at least several years to come.

This Is a Heading 1 Paragraph A

This is a paragraph of regular text. B

What Is CSS?

CSS is the abbreviation for Cascading Style Sheets, a language used to format web pages written in HTML. CSS enables you to control the visual layout and appearance of web pages, including the fonts, colors, spacing, and positioning used for text and other elements.

CSS consists of text-based instructions that specify the formatting to apply to particular elements. For

example, you could create an `h1` style to format the `h1` element mentioned in the previous section.

You can implement CSS in three ways: as an external file, as styles embedded in the HTML document, or as styles applied inline within a particular HTML tag. Using an external file is usually best, because it enables you to format multiple HTML documents using a single style sheet. When you need to make changes, you can change the external CSS rather than having to change the individual documents.

How Do You Create HTML and CSS?

Both HTML and CSS consist of text-only files, so you can create them using even the most basic text editor, such as the Notepad text editor included with Windows. However, to create HTML and CSS quickly and accurately, you will usually do better to use a text editor or integrated development environment that provides features for entering and checking code. Such text editors are often referred to as code editors.

This book recommends Microsoft's Visual Studio Code, a free code editor that runs on Windows, macOS, and Linux and that includes some integrated development environment features. See the section "Install Visual Studio Code," later in this chapter, for instructions on getting Visual Studio Code.

What Is Responsive Web Design?

In the early days of the Web, most people browsing it would use a desktop computer or laptop computer with a screen capable of displaying at least a moderate amount of information — say, 1024×768 resolution or higher. Most web pages were designed and coded to be easily readable on such screens. If you accessed such a web page using a much smaller or lower-resolution screen, you would likely see only part of the page's width at a readable size and would need to scroll horizontally to see the rest.

Nowadays, visitors use many different types of devices, from desktop computers with huge screens all the way down to tablets and smartphones with comparatively tiny screens. This variety of browsing devices means that one-size-fits-all web design is no longer satisfactory for most websites.

To cater to different devices, website builders use an approach called *responsive web design*. Responsive web design creates pages that can adapt to different device types, different screen sizes, and changes in orientation between portrait and landscape.

In responsive web design, a web page's layout and content automatically adjust to suit the screen size of the browsing device. Responsive web design uses flexible grid systems, images, and typography to change a web page's layout. It uses media queries to apply different styles suited to the device's screen size.

Responsive web design has several clear advantages over static web design. First, a responsive web page delivers a consistent user experience across different types of devices rather than favoring some devices over others. Second, a responsive web page is easier for visitors to read, navigate, and use. Third, a responsive web page improves accessibility, enabling people with disabilities to access it satisfactorily. Fourth, a responsive web page can improve search engine optimization, or SEO for short.

Understanding Static and Dynamic Web Pages

For your website, you can create either static web pages or dynamic web pages. A *static* web page is one whose content is fixed and does not change unless the page is edited. By contrast, a *dynamic* web page is one whose content changes as needed.

Static web pages are well suited to some purposes, and you will likely want to create some static pages for your website. However, it is likely that many of your web pages will benefit from displaying up-to-date information or from responding to a visitor's needs, so you will need to create dynamic pages, too.

Comparing Static Web Pages with Dynamic Web Pages

Static web pages are straightforward to create using HTML and CSS, the technologies on which this book focuses. Some static web pages may also benefit from functionality using the JavaScript scripting language.

Static web pages are suitable for websites that do not need frequent updates or content changes, such as company websites, landing pages, and personal blogs. Static web pages are also more secure than dynamic web pages, because they do not have a database connection that hackers might be able to exploit. Static pages may have a fixed format, but they can also be responsive, using media queries — discussed in Chapter 10 — to adapt to the screen of the device requesting them.

Given a fast Internet connection, static web pages should load quickly for visitors, because the server needs only to provide the existing file. By contrast, dynamic web pages typically require the server to perform some processing before it can send the web page to the browser.

Dynamic web pages are more complex than static web pages and take more work to create. Dynamic web pages require the use of server-side scripting languages such as PHP, ASP.NET, and Java.

Dynamic web pages enable you to create more interactive and feature-rich websites that can be updated frequently. Dynamic web pages are great for websites that benefit from frequent updating, such as news sites, social media sites, or e-commerce sites. Dynamic web pages give you greater flexibility than static web pages, because you can customize them to meet the needs of your company or organization.

As an example of the difference between static web pages and dynamic web pages, consider a web page that displays the menu for a restaurant. If you create a static page, the menu remains the same unless you edit the file. That is doable, but you might need to change the menu every day, updating the dishes and the prices. Instead, you could create a dynamic web page that pulls in the details of the day's special dishes from a database, together with the current price for each menu item. This way, the menu remains current without you needing to edit it.

What Is a Responsive Website?

A *responsive* website is one built to adapt automatically to different screen sizes and resolutions so as to provide a good viewing experience on all devices. Your website is likely to attract visitors using desktop computers, laptop computers, tablets, and smartphones, so you should make sure that your website appears in a satisfactory way on different screen sizes, resolutions, and aspect ratios.

A responsive website uses a CSS feature called *media queries* to determine the screen size and resolution of visiting devices and to adjust the layout, font sizes, and image sizes to suit the devices.

Comparing Responsive Websites and Nonresponsive Websites

A responsive website is a website that checks what type of device is accessing the site and displays its contents in a suitable way for that device. For example, if you visit a responsive website using your desktop computer, which has a large screen, the website serves your computer versions of the pages formatted for the large screen.

By contrast, if you go back to the same responsive website using your mobile phone, the web server serves up versions of the pages formatted to suit the smaller screen.

A nonresponsive website simply gives each visitor the same type of page, regardless of whether it fits the visiting device or not. The nonresponsive website does not check to see what type of device is visiting.

Normally, you would want to create a responsive website rather than a nonresponsive website. Building a responsive website has several key advantages:

- The website's content is consistently usable across different devices using a single codebase. You do not need to create separate websites for different types of devices.
- Having a single codebase simplifies developing and updating the website and reduces maintenance costs.
- Having the website viewable and usable on different devices can increase brand recognition and increases the likelihood of visitors sharing your website on social media, which may drive extra traffic to the website. Such success naturally also depends on the quality of your website's content; responsiveness helps, but it is no panacea.

See Chapter 10 for information on making your website responsive.

Understanding Tools for Creating Web Pages

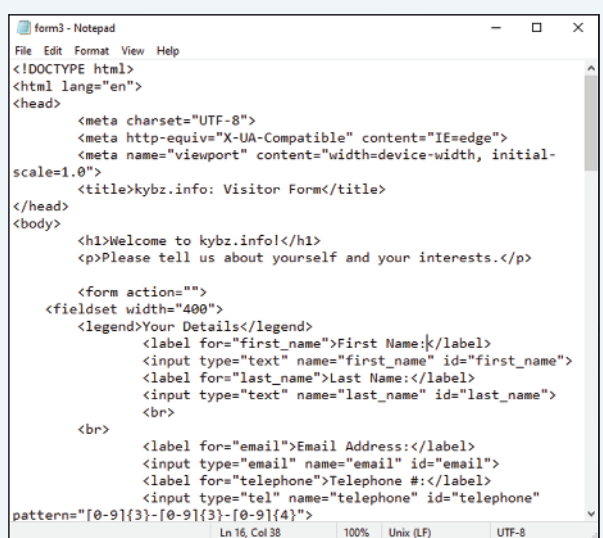
Many different types of tools are available for creating web pages and websites. This section summarizes the various types of tools available and then points you toward the tools this book uses to illustrate creating HTML and CSS files.

Both HTML files and CSS files contain only text, so you can create these files by using a text editor. However, you will likely prefer to use a code editor, an app that helps you enter code correctly and quickly. You will probably also need a graphics-manipulation app for creating images suitable for use on web pages.

Text Editors

A *text editor* is an app for creating and editing text. Both HTML files and CSS files consist only of text, so you can use even the most rudimentary text editor to create and edit them. For example, Windows includes the venerable but still serviceable text editor Notepad, shown editing an HTML file in the nearby illustration.

Notepad and other text editors offer no specific features for creating HTML and CSS. Some purists prefer this type of minimalist approach, but most people benefit from having help in completing and checking code.



```
form3 - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>kybz.info: Visitor Form</title>
</head>
<body>
  <h1>Welcome to kyzb.info</h1>
  <p>Please tell us about yourself and your interests.</p>

  <form action="">
    <fieldset width="400">
      <legend>Your Details</legend>
      <label for="first_name">First Name:</label>
      <input type="text" name="first_name" id="first_name">
      <label for="last_name">Last Name:</label>
      <input type="text" name="last_name" id="last_name">
      <br>
      <label for="email">Email Address:</label>
      <input type="email" name="email" id="email">
      <label for="telephone">Telephone #:</label>
      <input type="tel" name="telephone" id="telephone"
pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}">
    </fieldset>
  </form>
</body>
Ln 16, Col 38      100%  Unix (LF)  UTF-8
```

Word Processors

A *word processor* is an app for creating documents consisting of text, graphics, and other objects, laid out and formatted as needed. While you *can* use a word processor to create HTML files and CSS files, it is not usually a good choice, as it brings a plethora of features that you must avoid using, such as formatting, layout, graphical objects, and revision marking.

Where a word processor may be helpful is for creating web pages from your existing word processing documents. For example, Microsoft Word enables you to save documents to three web formats: the Single File Web Page format; the Web Page format; or the Web Page, Filtered format. Normally, you would choose the Web Page, Filtered format, because it gives the most compact result, retaining only the information needed to display the web page and discarding information relevant only to the document in Word format. The Web Page format saves all the Word formatting information as well, effectively saving the entire Word document in HTML format. The Single File Web Page format creates a large file containing all the objects required to make up the Word document.

Prepare to Create Your Website

Before creating your website, you may need to choose a web host on which to host the website, register a domain name under which the website will appear on the Web, and get and apply a Secure Sockets Layer, SSL, certificate to secure the traffic between your website and its visitors.

Which steps you will need to take depends on your situation. If you or your company already have web hosting, skip that step; likewise, skip the domain name and SSL certificate steps if you already have those. When ready, move on to the next section, “Install Visual Studio Code.”

Choose a Web Host

If you or your company do not have a web host, start by identifying a suitable one and signing up for a hosting plan appropriate to your needs.

Many web hosts are available, as you can find in seconds by searching on the Web. When evaluating web hosts, you will normally want to consider the following features:

- **Price.** Use price to select a range of web hosts and plans that you can afford, and then apply the other factors in this list to grade the hosts and plans. Do not judge on price alone in isolation.
- **Uptime and reliability.** Your website needs to be up, running, and available 24/7 to serve visitors. Choose a web host that offers a high percentage of uptime — 99.9 percent uptime is considered the minimum uptime percentage for dedicated hosts — and high reliability.
- **Customer support.** Make sure the web host offers strong customer support via all the channels you will want to use — email support, phone support, and live chat support.
- **Performance and speed.** Web users easily become frustrated with sites that are slow to load, so make sure your web host delivers fast loading speeds. Look for a web host that uses a content delivery network, CDN for short. A CDN is a geographically distributed server system that delivers web content to visitors based on their geographical location rather than delivering all content from a central point that may be geographically distant from some visitors.
- **Scalability.** Make sure the web host enables you to upgrade your hosting plan as your website and its traffic grow. Such scalability helps you avoid outgrowing your web host and having to move to another host, which is a major and expensive upheaval.
- **Security, backup, and recovery.** The web host should provide SSL certificates, malware detection, and firewalls to keep websites secure. The host should also offer set-and-forget backup features to keep your website’s data protected in case of corruption or hardware failure, plus easy-to-use tools for recovering your website from the latest viable backup.

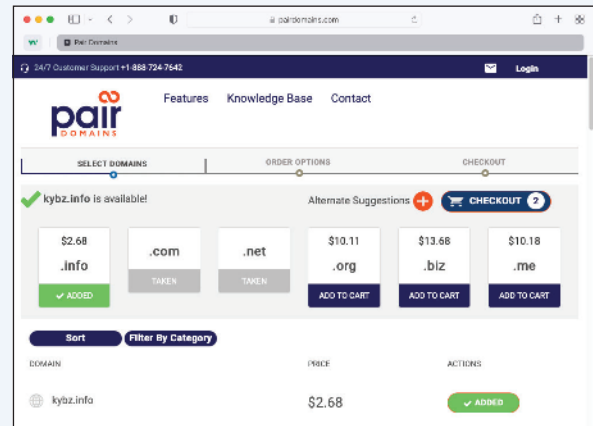
Register a Domain Name

If you do not have a domain name for your website, now is the time to get one. Open a browser window to a domain registrar, search to identify an available domain name that suits you, and register it. The nearby illustration shows the registration interface at Pair Domains.

As of this writing, these are five of the leading domain registrars:

- GoDaddy, www.godaddy.com
- Domain.com, www.domain.com
- Namecheap, www.namecheap.com
- Google Domains, <https://domains.google>
- Porkbun, www.porkbun.com

All these domain registrars offer a wide range of top-level domains, or TLDs. These TLDs range from .com, .org, and .net — three of the original six TLDs created in the 1980s — to newer TLDs such as .art, .biz, and .shop. Prices vary wildly, with the most popular TLDs being far more expensive.



Various TLDs are restricted to bodies that meet qualification criteria. For example, the .gov TLD is reserved for U.S. government agencies and entities, the .mil TLD is reserved for the U.S. military, and the .edu TLD is reserved for accredited post-secondary education institutions in the United States. Disappointingly, the .cat TLD is restricted to the Catalan linguistic and cultural community, but the .dog TLD, the .pet TLD, and the .animal TLD are open to all.

Choose a Type of SSL Certificate

SSL is the abbreviation for Secure Sockets Layer, a networking security protocol used to establish an encrypted link between a web browser and a web server, ensuring that all data passed between them remains private and secure even if it is intercepted in transit. To make sure that browsers can access your website safely, you will need to get an SSL certificate and apply it to the website's domain.

You have two main options for getting an SSL certificate. First, you can get an SSL certificate from your domain registrar when you register the website's name. Second, many web hosts offer SSL certificates for the domains you host on their servers. A third option is to get an SSL certificate from a different domain registrar, but this circuitous approach is seldom beneficial.

Usually, you would want to find out what SSL certificates your web host offers before paying for an SSL certificate from your domain registrar.

Various types of SSL certificates are available, such as the following:

- A *trial certificate* is a time-limited certificate that enables you to test whether the certificate meets your needs; if it does, you can buy another certificate to replace it.
- A *positive certificate* enables encryption for your website's data and has a relatively small relying party warranty, \$10,000.
- A *basic certificate* also enables encryption but has a much higher relying party warranty, \$250,000.
- A *positive wildcard certificate* enables encryption for multiple subdomains within your domain, so you do not need to buy a separate certificate for each subdomain.

Install Visual Studio Code

As explained in the section “Understanding Tools for Creating Web Pages,” earlier in this chapter, a code editor is your workaday tool for creating and editing HTML documents and CSS files. Many different code editors are available, but this book recommends Visual Studio Code, a powerful but free code editor from Microsoft. Visual Studio Code runs on Windows, macOS, and Linux, with a similar interface on each platform.

Microsoft offers two separate builds of Visual Studio Code. The Stable build is what you will normally want to install. The Insiders build contains new features and fixes and may not be entirely stable.

Install Visual Studio Code

- 1 In a web browser, go to code.visualstudio.com.

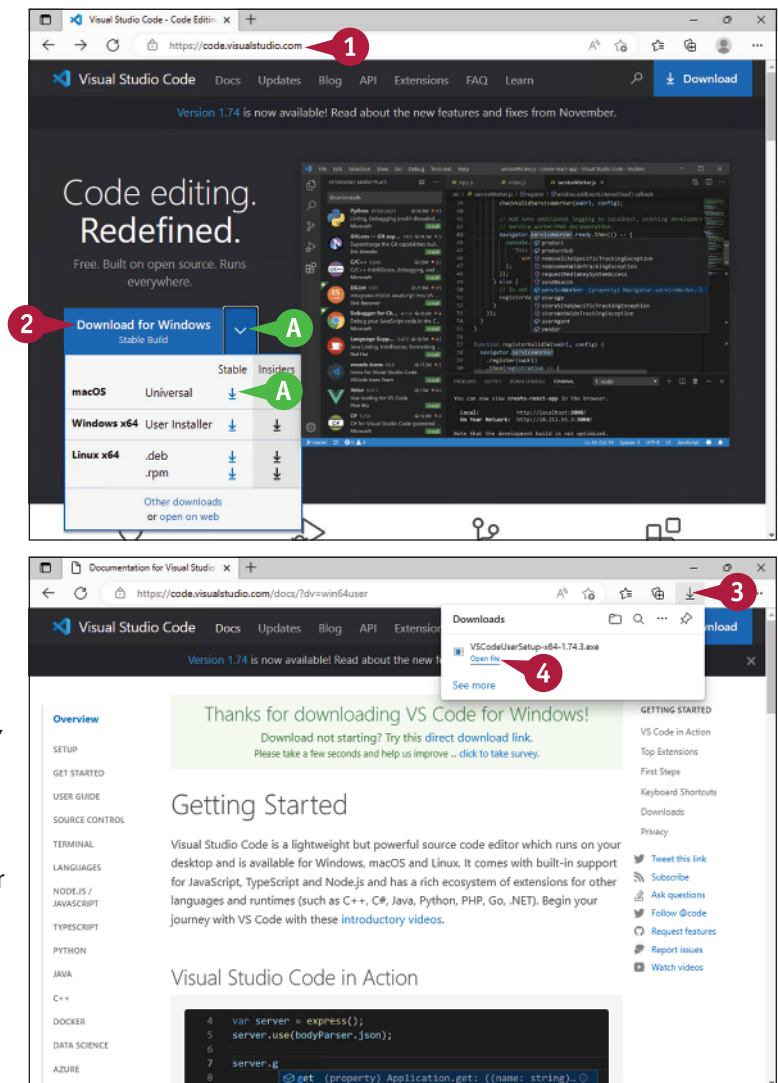
The Visual Studio Code website's home page appears.

- 2 Click **Download Stable Build**. This button shows your computer's operating system, which the page automatically detects.

- A If you want to download Visual Studio Code to use on a computer with a different operating system or if you want to download an Insiders build rather than a Stable build, click the drop-down arrow (▼), and then click **Download** (↓) for the operating system and build you want.

The download begins.

- 3 Open the browser's Downloads pane or window. For example, in Microsoft Edge, click **Downloads** (↓) if the Downloads pane does not open automatically.
- 4 Open the downloaded file. For example, in Microsoft Edge, click **Open file** under the downloaded file's name.



On Windows, the installer runs automatically.

- 5 On the License Agreement screen, click **I accept the agreement** (○ changes to ●) if you want to proceed with the installation.
- 6 Click **Next**.
- 7 On the Select Destination Location screen, click **Next** (not shown).
- 8 On the Select Start Menu Folder screen, click **Next** (not shown).

The Select Additional Tasks screen appears.

- 9 Select **Create a desktop icon** (☑) if you want to create a desktop icon for Visual Studio Code.
- 10 Select the two **Add “Open with Code” action** check boxes (☑) to give yourself an easy way to open files and folders in Visual Studio Code from File Explorer. See the second tip for details.

- 11 Select **Register Code as an editor for supported file types** (☑) to register Visual Studio Code with Windows as an app that can open file types such as HTML and CSS.

- 12 Select **Add to PATH (requires shell restart)** (☑) to add Visual Studio Code to the Windows path. This tells Windows where to find Visual Studio Code.

- 13 Click **Next**, and then click **Next** again on the Ready to Install screen.

The installation runs.

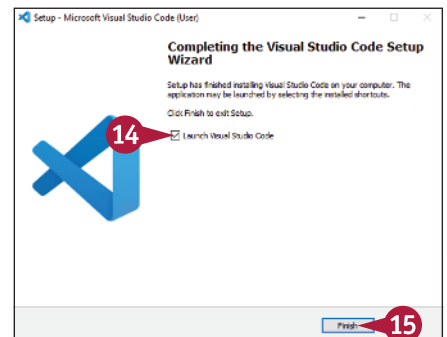
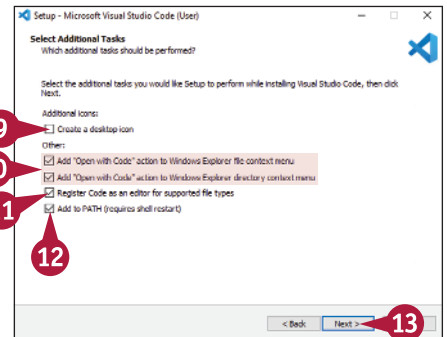
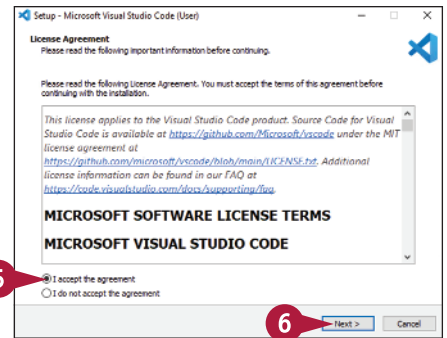
The Completing the Visual Studio Code Setup Wizard screen appears.

- 14 Click **Launch Visual Studio Code** (☑ changes to ☐) if you do not want to launch Visual Studio Code.

- 15 Click **Finish**.

The Setup Wizard closes.

Visual Studio Code opens, and you can configure it as explained in the next section.



TIPS

How do I install Visual Studio Code on macOS?

Double-click the downloaded Zip file to decompress it, and then drag the Visual Studio Code app file to the Applications folder.

What do the “Open with Code” options do?

Selecting **Add “Open with Code” action to Windows Explorer file context menu** (☑) enables you to open file types that Visual Studio Code supports by right-clicking them in File Explorer and then clicking **Open with Code** on the contextual menu. Similarly, selecting **Add “Open with Code” action to Windows Explorer directory context menu** (☑) enables you to open folders in Visual Studio Code.

Meet and Configure Visual Studio Code

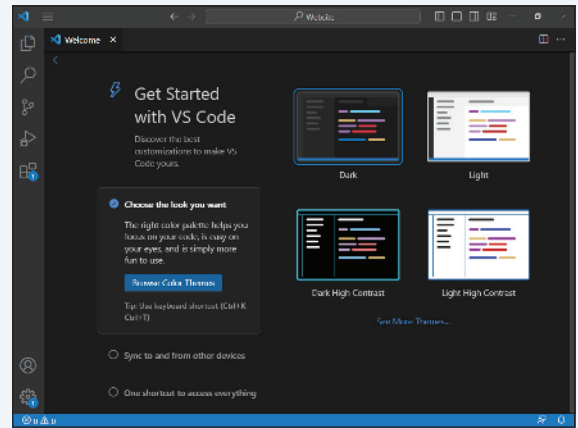
The first time you run Visual Studio Code, the app usually displays the Get Started with Visual Studio Code screen, which walks you through some initial configuration steps. You can return to the Get Started with Visual Studio Code screen later if you like; alternatively, you can use the app's other means of accessing its settings to configure the app to work the way you prefer.

The first change you will likely want to make is to the theme, which controls the overall look of Visual Studio Code. The app includes various dark themes and various light themes; third-party themes are also available.

Launch Visual Studio Code and Meet the Welcome Screen

Start by launching Visual Studio Code in the usual way for your computer's operating system. For example, on Windows, click **Start** (📁) to display the Start menu, and then click **Visual Studio Code** (🚀).

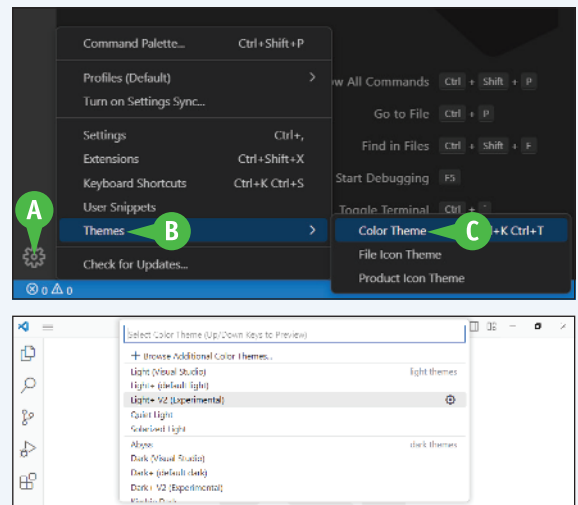
The first time you launch Visual Studio Code, the app automatically displays the Welcome screen, which encourages you to choose several key configuration settings. For example, you can click **Get Started with VS Code** to display the Get Started with VS Code screen, shown here, which provides links to several key settings. You can also configure these settings later, as explained in the following subsections.



Choose the Theme for Visual Studio Code

By default, Visual Studio Code uses its Dark+ theme, which is easy on the eyes in low-light conditions but tends to get over-inked in books. To change the theme, click **Manage** (A, ⚙️) in the lower-left corner to display the Manage pop-up menu, click **Themes** (B) to display the Themes continuation menu, and then click **Color Theme** (C), as shown here.

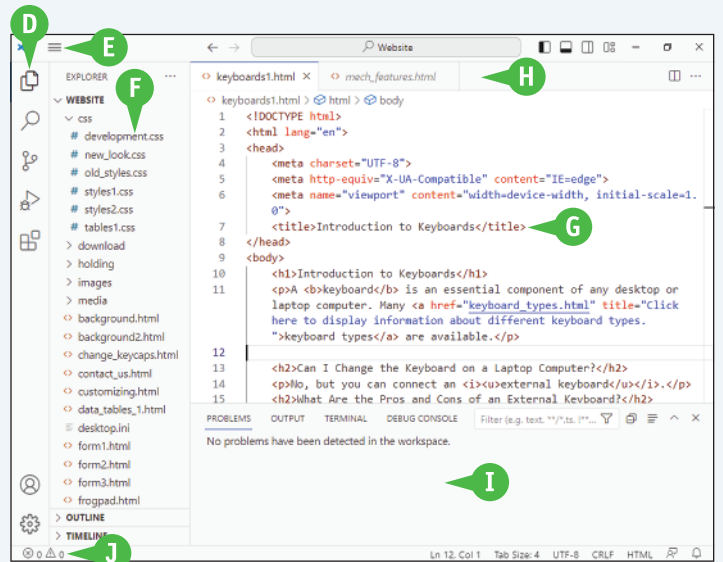
In the Color Theme picker, press **↑** and **↓** to move the selection highlight up and down the list of themes. Visual Studio Code displays a preview of the selected theme. When you settle on the theme you want, press **Enter** to apply it. From here on, this book uses the Light+ V2 (Experimental) theme for readability.



Identify the Key Elements of the Visual Studio Code Window

The Visual Studio Code window includes the following key elements:

- **Activity Bar (D).** This vertical bar on the left side of the window provides quick access to several different panels that you can display one at a time in the Primary Side Bar.
- **Menu icon (E).** When the menu bar is not displayed, click **Menu** (\equiv) to show the menu list, and then click the menu you want to open.
- **Primary Side Bar (F).** This area, which appears on the left side of the window by default, displays any of several panels, such as the Explorer panel shown in the nearby illustration. You switch from one panel to another by clicking the buttons in the Activity Bar. You can display or hide the Primary Side Bar as needed.
- **Secondary Side Bar (not shown).** This area on the right side of the window can display additional information about the code you are editing. For example, you can display an outline of the code structure. You can display or hide the Secondary Side Bar, as needed.
- **Editor (G).** This main area is where you edit your files.
- **Tab bar (H).** The Tab Bar displays a tab for each open document. You can click the tab for the document you want to display.
- **Panel (I).** This area to the bottom of the window has tabs that enable you to display different items: Problems, Output, Terminal, and Debug Console. You can display or hide the Panel, as needed.
- **Status Bar (J).** This narrow horizontal bar across the bottom of the window shows information about the current state of the Editor, including the language mode, the line and column numbers, and the number of problems detected.



continued ▶

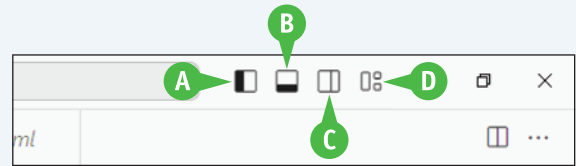
Meet and Configure Visual Studio Code (continued)





Visual Studio Code has a highly configurable interface that enables you to display only the elements you want to see at any particular time. The Activity Bar lets you display the panel you want to use in the Primary Side Bar, but you can hide the Primary Side Bar when you want to concentrate on your code in the Editor.

Visual Studio Code offers many preferences that you can set to control the way the app works. This section shows you how to set two of the most important preferences, Auto Save and Font Size.



Change the Layout of the Visual Studio Code Window

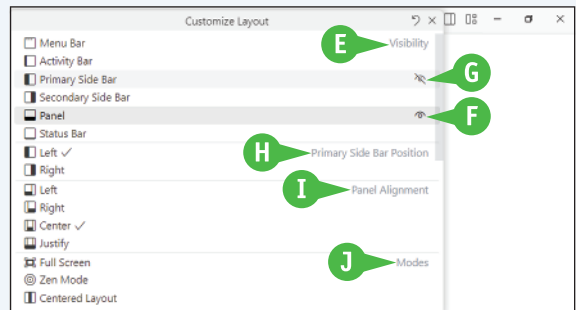
The four buttons in the upper-right corner of the Visual Studio Code window enable you to adjust the app's layout:




- **Toggle Primary Side Bar (A, )**. Click this button to toggle the display of the Primary Side Bar. Alternatively, press **Control** + **B** on Windows or Linux or **⌘** + **B** on the Mac.
- **Toggle Panel (B, )**. Click this button to toggle the display of the Panel. Alternatively, press **Control** + **J** on Windows or Linux or **⌘** + **J** on the Mac.
- **Toggle Secondary Side Bar (C, )**. Click this button to toggle the display of the Secondary Side Bar. Alternatively, press **Control** + **Alt** + **B** on Windows or Linux or **⌘** + **Option** + **B** on the Mac.
- **Customize Layout (D, )**. Click this button to display the Customize Layout pane, shown nearby.




The Customize Layout pane contains four sections: the Visibility section, the Primary Side Bar Position section, the Panel Alignment section, and the Modes section.


In the Visibility section (E), you can choose which elements to display and which to hide. Move the pointer over the element you want to display or hide. The Displayed icon (F, ) or the Hidden icon (G, ) appears to the right of the element. Click this icon to toggle the element between displayed and hidden.



In the Primary Side Bar Position section (H), click **Left** () or **Right** () to control which side the Primary Side Bar appears. The default is Left.

In the Panel Alignment section (I), click **Left** (), **Right** (), **Center** (), or **Justify** (), as needed.

In the Modes section (J), click **Full Screen** () to switch to Full Screen Mode; click **Zen** () to switch to Zen Mode, which strips down the Visual Studio Code interface to a minimum; or click **Centered Layout** () to switch to Centered Layout Mode. You can apply any or all of these three modes.

When you finish choosing options in the Customize Layout pane, click **Close** () to close the pane.

Configure Auto Save and Font Size

In Visual Studio Code, click **Menu** (K, ≡) to display the menu list, and then click **File** to open the File menu. Click **Preferences** to open the Preferences submenu, and then click **Settings**. The Settings screen appears.

The Commonly Used settings category normally appears at first. If another category appears, click **Commonly Used** (L).

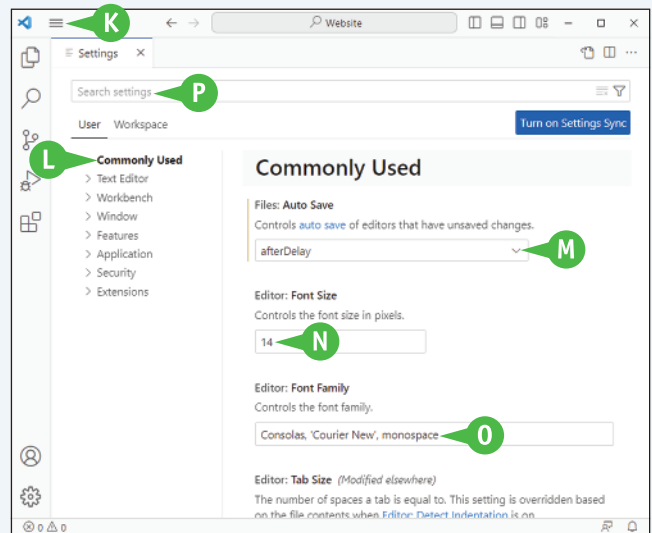
Click **Files: Auto Save** (M, ▼), and then click **afterDelay**. This setting makes Visual Studio Code save you changes automatically after a short delay. This book's instructions assume that you are using the `afterDelay` setting. The other available settings are `off`; `onFocusChange`, which saves changes when you move the focus to a different part of Visual Studio Code; and `onWindowChange`, which saves changes when you move the focus to a different window.

Click **Editor: Font Size** (N) and type the font size you want to use in the editor.

Optionally, click **Editor: Font Family** (O) and type the font family you want to use in the editor. Visual Studio Code uses the first font — in the example, Consolas — if it is available, falling back to subsequent named fonts if necessary; if none of the named fonts is available, it falls back to the generic font family — in this case, monospace. If the font name includes a space, enter the font in quotes, as in 'Courier New' in the example.

As you can see, Visual Studio Code has a huge number of features. You can browse through them by clicking the categories in the left pane or search for specific settings by clicking **Search settings** (P) and typing your search term.

When you finish configuring settings, click **Close** (X) to close the Settings screen.



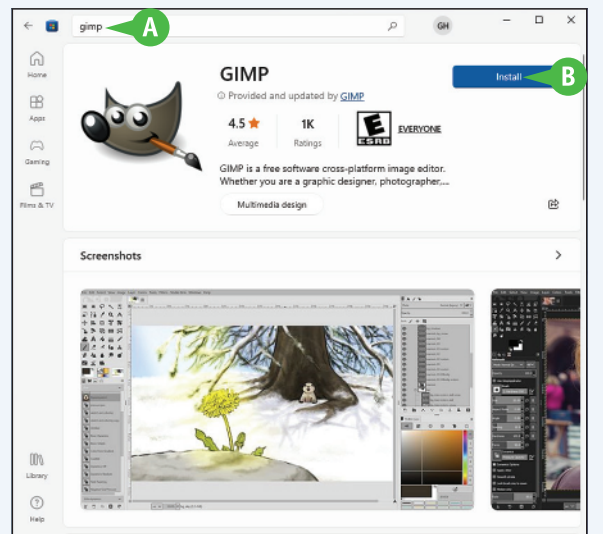
Install GIMP

To create suitable image files for your website, you will need an image editor app. Many apps are available, some free and some not, so you have plenty of choices. The best cross-platform solution is GIMP, the GNU Image Manipulation Program. GIMP is a free and open-source image editor that runs on Windows, macOS, Linux, and other operating systems.

Install GIMP on Windows 10 or Windows 11

To install GIMP on Windows 10 or Windows 11, click **Start** (🏠) to display the Start menu, and then click **Microsoft Store** (🛒) to open the Microsoft Store app. Click **Search apps, games, movies, and more** at the top of the app window, type **gimp** (A), and then press **Enter**. In the search results, click **GIMP**, and then click **Install** (B).

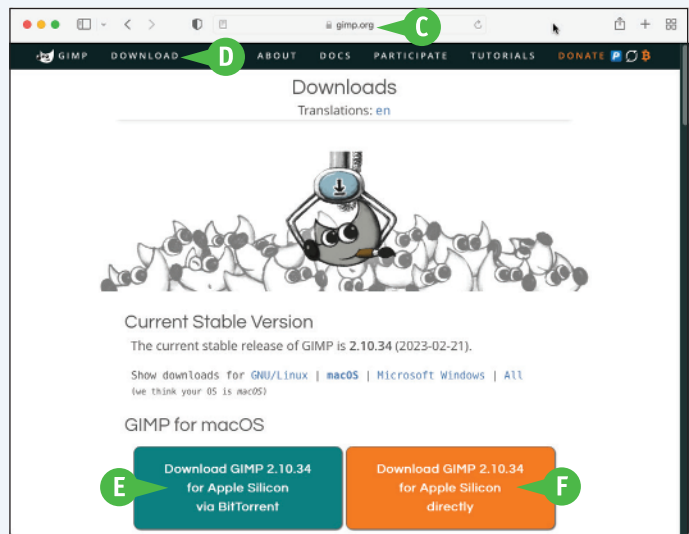
Once GIMP is installed, you can launch it by clicking **Start** (🏠) and then clicking **GIMP** (🐼).



Install GIMP on macOS

As of this writing, GIMP is not available on Apple's Mac App Store.

To install GIMP on macOS, open a browser window or tab to the GIMP website, www.gimp.org (C). Click the **Download** tab (D) at the top of the window to display the Downloads page, locate the current stable version for macOS, and then click either **Download GIMP via BitTorrent** (E) to download GIMP via the BitTorrent file-sharing service or **Download GIMP Directly** (F) to download GIMP from the GIMP website. The buttons offer you the choice of downloading for Apple Silicon, Apple's home-grown M-series processors, or for Intel, the processors used by older Macs; choose appropriately.



If the Do You Want to Allow Downloads on www.gimp.org? dialog box opens, click **Allow**.

When the download finishes, click **Show Downloads** (⌘D), and then click the GIMP disk image file in the Downloads list. A Finder window opens showing the contents of the GIMP Install package. Drag the GIMP icon (G) to the Applications folder shortcut.

After installation finishes, click **View** to open the View menu, and then click **Show Toolbar** to display the toolbar and the sidebar. Go to the Locations section of the sidebar and click **Eject** (⏏) to eject the GIMP disk image file. You can then delete the file from the Downloads folder if you do not want to keep it.

With the app installed, you can run GIMP by clicking **Launchpad** (⌘Space) to display the Launchpad screen and then clicking **GIMP** (🐼).



Install GIMP on Linux

You have several options for installing GIMP on Linux, including installing it via your distribution's software-management app.

If your computer is running Ubuntu, the easiest approach is to open a terminal window and update the package list by issuing the following command:

```
sudo apt-get update
```

Once the package list is up to date, install GIMP and all necessary dependencies by issuing the following command:

```
sudo apt-get install gimp
```

When installation finishes, you can launch GIMP either by opening the application menu and clicking **GIMP** (🐼) or by issuing the `gimp` command in a terminal window:

```
gimp
```

Install the Major Browsers

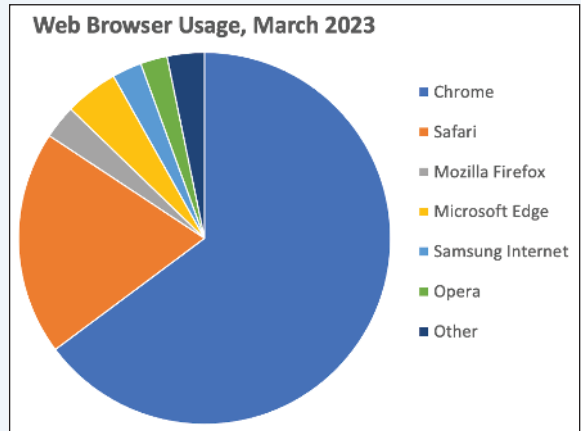
Most operating systems come with a single browser app — for example, Windows comes with the Microsoft Edge browser, macOS includes Apple’s Safari browser, and ChromeOS features Google’s Chrome browser. But while each of these browsers works fine for web browsing, you will likely want to install a full deck of major browsers so that you can test your web pages with all of them and work through any compatibility problems that arise.

As of this writing, the four leading browsers are Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

Which Browsers Are Most Widely Used?

A wide range of web browsers are available, but as of this writing in spring 2023, Google’s Chrome dominates the market. Breakdowns of web browser usage (see the nearby pie chart) tend to be illustrative rather than exact, but most figures agree that Chrome has between two-thirds and three-quarters of traffic. Next comes Apple’s Safari with approximately one-fifth of traffic, mostly thanks to its inclusion on every iPhone, every iPad, and every Mac.

After Safari is Microsoft Edge, included with Windows 10 and Windows 11, with a little less than 5 percent of traffic. Mozilla Firefox has around 3 percent; Samsung Internet has around 2.5 percent, as does Opera. Other browsers, such as Vivaldi and the now-terminated Internet Explorer, make up the remaining 3 percent of traffic.



How Do Chrome, Safari, Mozilla Firefox, and Microsoft Edge Compare to Each Other?

All four browsers support a wide range of features that let users browse the web efficiently and enjoy web content. Each browser supports HTML5, CSS3, and JavaScript. Each offers the following seven key features:

- **Tabbed browsing.** You can open multiple web pages on separate tabs within a single window.
- **Bookmarks.** You can mark web pages so that you can quickly visit them again.
- **History.** The browser automatically keeps a list of pages you visit, enabling you to return to them easily.
- **Private or “Incognito” browsing.** This mode lets you turn off history when you do not want the browser to track it.
- **Autofill.** The browser stores data you enter in form fields — for example, address data — so that it can fill in those fields automatically in future forms.
- **Password storage.** The browser enables you to store passwords securely so that you can enter them in web pages without typing.

- **Extensions.** You can extend the browser’s functionality by installing extensions, third-party add-ons. Beyond these features, which are arguably essential for browsing the web nowadays, each browser offers features of its own. For example:
 - Chrome is tightly integrated with Google services such as Gmail and Google Drive, and it offers Chrome apps, web apps that run by using the Chrome browser.
 - Microsoft Edge offers tight integration with Office 365 and other Microsoft services.
 - Mozilla Firefox offers a highly customizable interface and strong privacy features.
 - Safari integrates tightly with Apple device features such as Continuity, which lets you start browsing on one device and continue on another, and Apple’s Keychain password manager. Safari also offers a Reader Mode to reduce ads and interruptions on web pages.

Which Browsers Should You Install?

Unless you have strong reasons to do otherwise, you should probably install all four of the leading browsers — Chrome, Safari, Mozilla Firefox, and Microsoft Edge — on your development computer. Having all four browsers available will enable you to make sure that your web pages display correctly on the vast majority of computers.

Install the Browsers and Choosing Configuration Options

You can download the browsers from the following sites:

- Chrome: www.google.com/chrome
- Safari: <https://support.apple.com/downloads/safari>
- Firefox: www.mozilla.org/en-US
- Edge: www.microsoft.com/edge/download

Installing the browsers is straightforward enough, but you will typically need to make the following three choices:

- **Whether to make the new browser the default browser for your computer.** For example, in the Make Firefox Your Go-To Browser dialog box, you would click **Skip this step** to turn down the offer to make Firefox the default browser.
- **Whether to import bookmarks from your current browser.** Importing bookmarks is handy when you are switching permanently or semipermanently from one browser to another, but when installing multiple browsers for use in parallel, you may want to keep separate bookmarks.
- **Whether to sync your browser data across your devices.** Syncing bookmarks, passwords, and settings data across all the devices on which you use a particular browser can be a great boon for consumers. But when you are installing the browsers on your computer for testing, you may prefer to keep the data unsynced.

Create a Folder Structure for Your Website

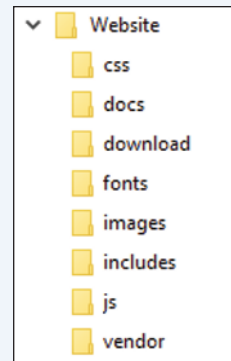
Part of the planning for your website should be choosing and implementing a folder structure for the website. Normally, you would set up this folder structure both on your development computer — or development server — and on the live web server so that you can copy files from your development environment to your live environment without having to change the relative paths to the folders.

There is no official “approved” folder structure for websites, so you can arrange folders as best suits you and your colleagues. This section suggests approaches to get you started.

List the File Types Your Website Will Use

Start by making a list of the different types of files your website will use. The following types are widely used, but your website may well have others:

- HTML documents and CSS files
- Images
- Fonts
- Audio files and video files
- JavaScript script files
- Documentation files about the website
- Reusable code and templates



Identify the Directories Your Website Will Need

Next, decide where you will store these items. Table 1-1 and the nearby illustration show a fairly typical list of directories, but you will likely want to create your own custom version.

Table 1-1: Typical Directories in a Website’s Structure

Folder	Explanation
/	The root directory, the top level of the website. You will put the main <code>index.html</code> file, your web pages, and other essential files, such as a <code>robots.txt</code> file, in this directory. The other directories are children of the root directory.
/css	The directory for storing the CSS files you use to style your web pages. If you have many CSS files, consider creating subdirectories to organize them more tightly.
/docs	The directory for storing documentation files.
/download	The directory for storing files your website makes available for download.
/fonts	The directory for storing font files the website uses.
/images	The directory for storing image files.
/includes	The directory for storing templates or reusable code.
/js	The directory for storing JavaScript script files.
/vendor	The directory for storing third-party frameworks or libraries your website uses. See Chapter 12.

Adapt the List to Suit Your Needs

Looking at this list, you might ask where the audio files and video files go in the directory structure. As the directory structure makes no provision for these file types, you will need to decide on a suitable place.

You could simply create an `/audio` directory for audio files and a `/videos` directory for video files. But to keep the directory structure compact, you might prefer to create a directory such as `/media` or `/assets` to contain subdirectories for audio, video, and perhaps images. For example:

```
/media
  /audio
  /images
  /videos
```

Or:

```
/assets
  /audio
  /images
  /videos
```

When deciding on a location for your video files, keep in mind that video's hefty file sizes will increase the load on your web server. If your website serves many videos to visitors, consider offloading the burden by storing the video files on a CDN.

Create the Directory Structure

Once you have made the design decisions, you can create the directory structure using standard commands for your computer's operating system. For example:

- **Windows.** Open a File Explorer window showing the folder in which you will place the root directory. Right-click open space in the folder, click or highlight **New** on the contextual menu, and then click **Folder** on the continuation menu. Type the directory name and press **Enter** to apply the name. Press **Enter** again to open the directory. You can then create subdirectories inside it.
- **macOS.** Open a Finder window to the folder in which you will place the root directory. Press **Control**+click or right-click to display the contextual menu, and then click **New Folder**. Type the directory name and press **Return** to apply it. Double-click the directory to open it so that you can create subdirectories inside it.
- **Linux.** Open a Terminal window, use the `cd` command to navigate to the appropriate directory, and then use the `mkdir` command to create the root directory. Use the `chdir` command to change directory to the root directory, and then create subdirectories inside it.