

Teach Yourself
VISUALLY[™]

HTML and CSS

2nd Edition

Guy Hart-Davis



Teach Yourself
VISUALLY[™]

HTML and CSS

2nd Edition

by Guy Hart-Davis



Teach Yourself VISUALLY™ HTML and CSS

2nd Edition

Copyright © 2023 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada and the United Kingdom.

ISBNs: 9781394160686 (paperback), 9781394160693 (ePDF),
9781394160709 (ePub)

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at www.wiley.com/go/permission.

Trademarks: Wiley, the Wiley logo, Visual, the Visual logo, Teach Yourself VISUALLY, Read Less - Learn More, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THE INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

FOR PURPOSES OF ILLUSTRATING THE CONCEPTS AND TECHNIQUES DESCRIBED IN THIS BOOK, THE AUTHOR HAS CREATED VARIOUS NAMES, COMPANY NAMES, MAILING, E-MAIL AND INTERNET ADDRESSES, PHONE AND FAX NUMBERS AND SIMILAR INFORMATION, ALL OF WHICH ARE FICTITIOUS. ANY RESEMBLANCE OF THESE FICTITIOUS NAMES, ADDRESSES, PHONE AND FAX NUMBERS AND SIMILAR INFORMATION TO ANY ACTUAL PERSON, COMPANY AND/OR ORGANIZATION IS UNINTENTIONAL AND PURELY COINCIDENTAL.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

For technical support please visit <https://hub.wiley.com/community/support>.

If you believe you've found a mistake in this book, please bring it to our attention by emailing our reader support team at wileysupport@wiley.com with the subject line "Possible Book Errata Submission."

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Control Number: 2023938158

Cover images: © Svetlana Ivanova/Getty Images

Cover design: Wiley

About the Author

Guy Hart-Davis is the author of more than 175 computer books, including *Teach Yourself VISUALLY iPhone 14*, *Teach Yourself VISUALLY MacBook Pro and MacBook Air*, *Teach Yourself VISUALLY Google Workspace*, *Teach Yourself VISUALLY Chromebook*, *Teach Yourself VISUALLY Word 2019*, *Teach Yourself VISUALLY iPad*, and *Teach Yourself VISUALLY Android Phones and Tablets*, 2nd Edition.

Author's Acknowledgments

My thanks go to the many people who turned my manuscript into the highly graphical book you are holding. In particular, I thank Jim Minatel for asking me to write the book; Lynn Northrup for managing the book's writing and technical review and improving the text; Kim Wimpsett for skillfully editing the text; Doug Holland for reviewing the book for technical accuracy and contributing helpful suggestions; Susan Hobbs for proofreading the book minutely; and Straive for laying out the book.



How to Use This Book

Who This Book Is For

This book is for the reader who has never used this particular technology or software application. It is also for readers who want to expand their knowledge.

The Conventions in This Book

1 Steps

This book uses a step-by-step format to guide you easily through each task. **Numbered steps** are actions you must do; **bulleted steps** clarify a point, step, or optional feature; and **indented steps** give you the result.

2 Notes

Notes give additional information — special conditions that may occur during an operation, a situation that you want to avoid, or a cross reference to a related area of the book.

3 Icons and Buttons

Icons and buttons show you exactly what you need to click to perform a step.

4 Tips

Tips offer additional information, including warnings and shortcuts.

5 Bold

Bold type shows command names, options, and text or numbers you must type.

6 Italics

Italic type introduces and defines a new term.

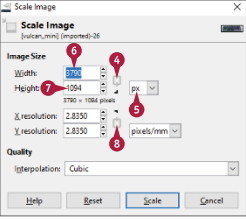

Resize an ImageCHAPTER 4
Including Images

GIMP enables you to resize an image to the dimensions you need, either increasing or decreasing the image's width and height. Given the high resolutions at which current digital cameras and smartphones shoot photos, you are more likely to need to decrease the image's dimensions when working with photos, but you may need to enlarge smaller graphics.

When resizing an image, you can choose what type of interpolation to use. *Interpolation* is how the app calculates the color values for pixels it inserts when resizing an image. You can also adjust the image's resolution if needed.

Resize an Image

- 1 Open the image file you want to resize.
- 2 **Note:** Normally, you would rotate, straighten, and crop an image, as needed, before resizing it.
 - 3 The image's resolution appears in the title bar.
- 1 Click **Image**.
The Image menu opens.
- 3 Click **Scale Image**.
The Scale Image dialog box opens.
- 4 If you want to adjust the width and height separately, click **Unlinked** (☐ changes to ☒).
- 5 Verify that **px** appears in this drop-down list. If it does not, click **▼**, and then click **pixels**.
- 6 Click **Width** and enter the width in pixels.
- 7 If you unlink the width and height, click **Height** and enter the height in pixels.
- 8 If you will adjust the resolution and want to adjust the horizontal and vertical resolution separately, click **Linked** (☒ changes to ☐).



- 9 Verify that **pixels/mm** appears in this drop-down list. If not, click **▼**, and then click **pixels/mm**.
- 10 Click **X resolution** and enter the horizontal resolution.
- 11 If you unlink the resolutions, click **Y resolution** and enter the vertical resolution.
- 12 Click **Interpolation** (▼), and then click the type of interpolation you want to use: **None**, **Cubic**, **Linear**, **NoHalo**, or **LoHalo**. See the tip for advice.
- 13 Click **Scale**.

The Scale Image dialog box closes.

- 4 GIMP resizes the image as you specified.
- 4 The image's adjusted resolution appears in the title bar.

You can now modify the image further, as needed; save it; and close it.

TIP

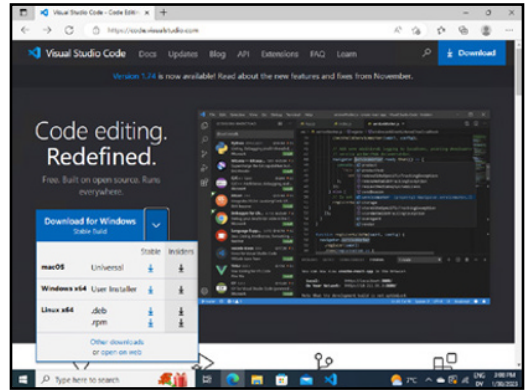
Which type of interpolation should I use when resizing an image? When you are reducing the image's size, choose **LoHalo** in the Interpolation drop-down list; if LoHalo delivers a disappointing result for that image, try **NoHalo** instead. When you are increasing the image's size, choose either **Cubic** or **Linear**. Cubic interpolation takes more processing power than linear interpolation, so it may take longer.

If you just need a rough-and-ready resized image, you can choose **None**, which performs no interpolation but instead copies the color of each pixel from the closest adjacent pixel in the original image. The resulting image may be grainy or coarse.

Table of Contents

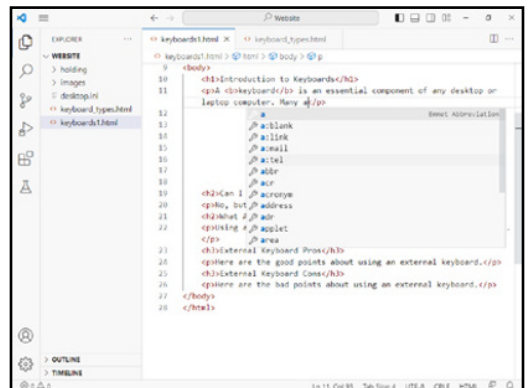
Chapter 1 Getting Ready to Create Websites

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



Chapter 2 Creating Your First Web Pages

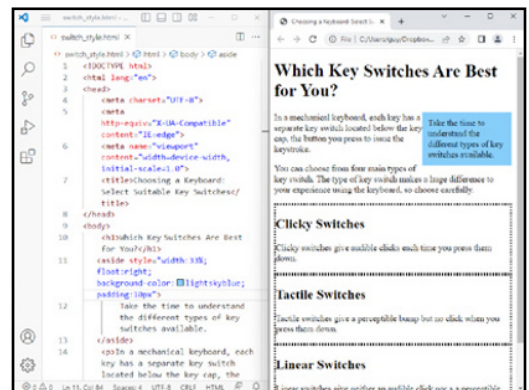
Study the Anatomy of a Web Page.....	28
Tell Visual Studio Code Which Folder to Use	30
Create Your First Web Page.....	32
Open the Web Page in a Browser	34
Add Headings and Text.....	36
Nest One Element Within Another Element.....	38
Add Comments	39
Apply Direct Formatting.....	40
View a Page's Source Code.....	41
Validate a Web Page.....	42



Create Another Web Page	44
Understanding the Essentials of Hyperlinks.....	45
Create a Hyperlink Between Your Web Pages	46
Interpret HTTP Status Codes	48

Chapter 3 Structuring a Web Page

Meet the Elements for Structuring Web Pages.....	52
Select Items with span and div Elements.....	54
Create header Elements and footer Elements	56
Add article Elements to a Page.....	58
Create Pull Quotes with the aside Element.....	60
Divide a Page Using section Elements.....	62
Create Collapsible Sections.....	64



Chapter 4 Including Images

Grasp the Essentials of Web Image Formats.....	68
Launch GIMP and Perform Essential Moves.....	70
Rotate or Straighten an Image.....	74
Crop an Image.....	76
Resize an Image	78
Reduce the Number of Colors in an Image	80
Convert an Image to the Format You Need.....	82
Learn the HTML for Images.....	84
Insert an Image.....	86
Create a Figure with a Caption	88

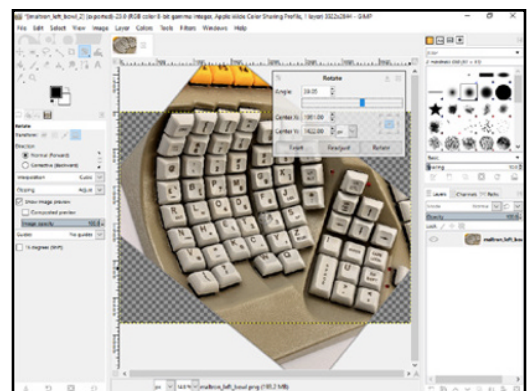
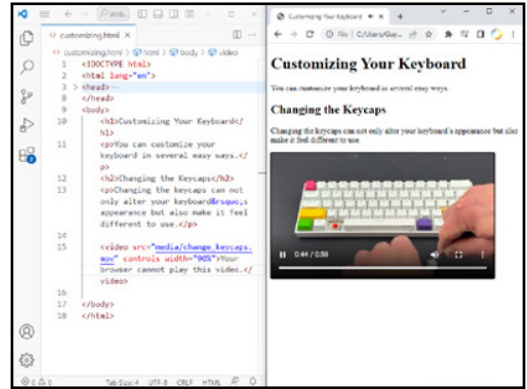


Table of Contents

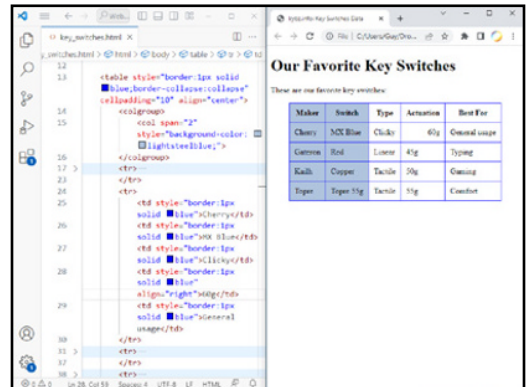
Chapter 5 Working with Links

Grasp the Essentials of Links	92
Create a Link to a Web Page	94
Create a Link to Elsewhere on the Same Web Page	96
Specify the ScreenTip for a Link	98
Redirect the Browser to a Different Page	99
Create a Link for Downloading a File	100
Create a Link That Starts an Email Message	101
Create a Link from an Image	102
Create Multiple Links from an Image	104
Include an Audio File in a Web Page	108
Include a Video File in a Web Page	110
Embed a YouTube Video in a Web Page	112



Chapter 6 Creating Lists and Tables

Grasp the Different Types of Lists	116
Create a Numbered List	118
Create a Bulleted List	120
Create a Definition List	122
Nest One List Inside Another List	124
Learn the HTML for Tables	126
Create a Table	128
Add Rows or Columns to a Table	130
Specify Table Width and Column Width	132
Format Table Borders	134
Adjust Table Padding and Spacing	136
Create Groups of Columns	138
Align Tables, Rows, and Cells	140
Create Cells That Span Rows or Columns	142

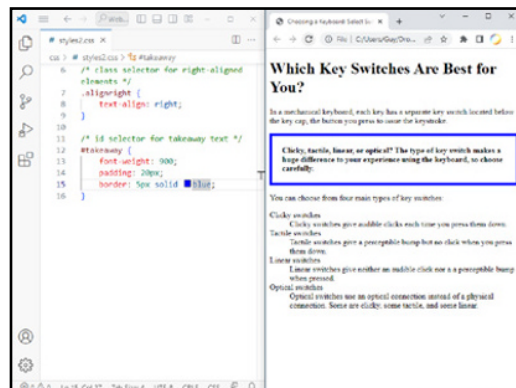


Set a Background Color or Image for a Table 143
 Nest One Table Inside Another Table..... 144

Chapter 7

Getting Started with CSS

Grasp How CSS Works..... 148
 Format Elements with Inline CSS 150
 Format a Page Using Internal CSS..... 152
 Create an External CSS File 154
 Link an External CSS File to a Web Page 156
 Distinguish Element, Class, and ID Selectors 158
 Apply Styles Using Element Selectors 159
 Apply Styles Using Class Selectors 160
 Apply Styles Using ID Selectors..... 162



Chapter 8

Formatting Text with CSS

Understanding Fonts and How to Use Them 166
 Specify the Font Family 168
 Set the Font Size and Font Weight..... 170
 Adjust Line Height and Letter Spacing 172
 Create Superscripts and Subscripts..... 174
 Understanding Ways to Set Color in CSS 176
 Set Font Color 177
 Apply Text Shadows 178
 Display Monospaced Font 179
 Apply Text Decoration 180
 Understanding HTML Entities..... 182
 Insert Special Characters with HTML Entities 184
 Insert Emojis..... 185
 Using Custom Fonts on Web Pages..... 186

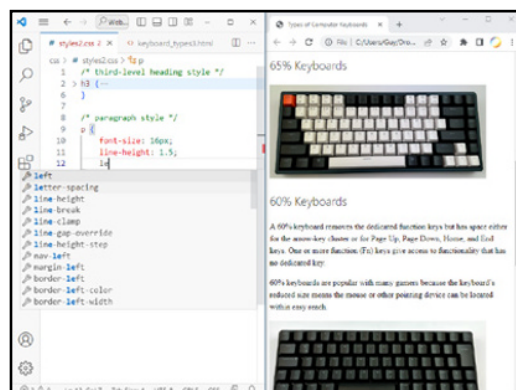
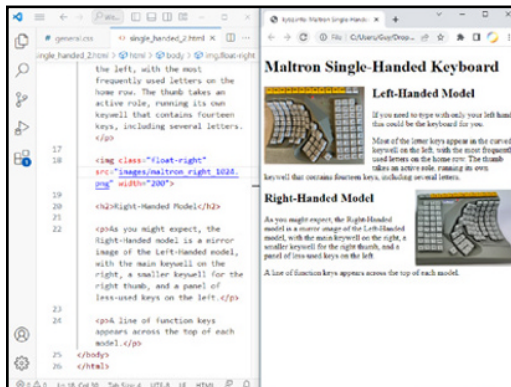


Table of Contents

Chapter 9 Sizing and Positioning with CSS

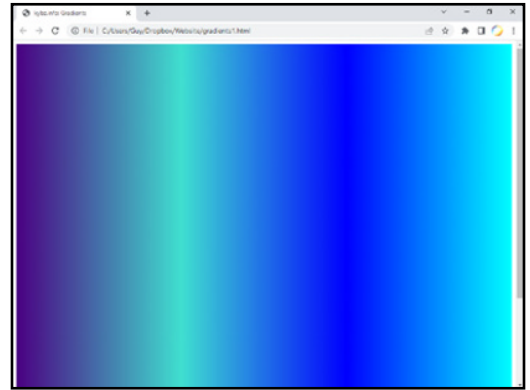
Understanding Pseudo-Classes	190
Apply Contextual Formatting with Pseudo-Classes	192
Understanding CSS Combinators	194
Target Elements Using CSS Combinators	196
Understanding Pseudo-Elements	198
Apply CSS to Pseudo-Elements	200
Override CSS by Using the !important Declaration	202
Understanding the CSS Box Model	204
Understanding Ways of Sizing Elements.....	206
Specify the Size for an Element.....	208
Specify Padding and Borders for an Element	210
Set Margins to Control Element Spacing	212
Understanding CSS Positioning Essentials.....	214
Create Block Quotes	216
Fix an Element in Place in the Viewport	218
Float an Element Beside Another Element.....	220
Understanding the display Property	222
Control the Display of an Element	223
Create a Flexbox Layout.....	224



Chapter 10

Creating Responsive and Appealing Pages

Understanding the Tools for Creating Responsive Pages	228
Apply Relative Sizing.....	230
Add Media Queries to a Page	232
Set Tap Targets for Touch Screens	234
Set the Viewport Size	236
Check Your Pages on Various Devices and Screens	238
Understanding How CSS Gradients Work	240
Apply a Linear Gradient to an Element	242
Apply a Radial Gradient to an Element	244
Using Sprites.....	246
Understanding CSS Animations	248
Apply Transitions to HTML Elements	250
Create a Keyframe Animation.....	252



Chapter 11

Creating Forms

Grasp Web Form Essentials	256
Create a Form.....	258
Add Text Input Controls to a Form.....	260
Add Radio Buttons to a Form.....	262
Add Check Boxes to a Form	264
Add a Drop-Down List of Options.....	266
Add Command Buttons to a Form	268

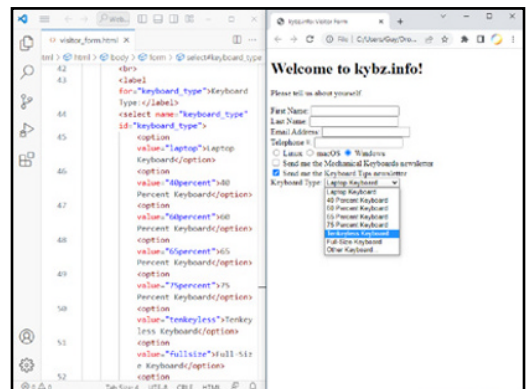
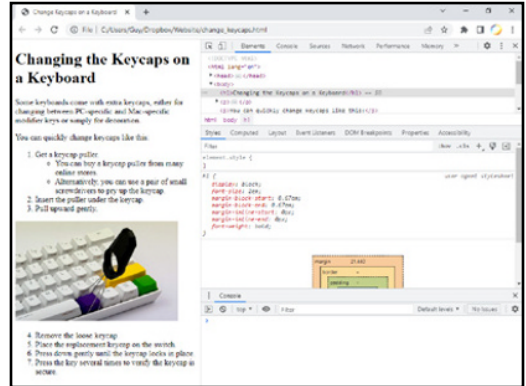


Table of Contents

Chapter 12 Taking Your Website to the Next Level

Understanding How Search Engines Work.....	272
Optimize Your Website for Search Engines	274
Guide Search Spiders with a robots.txt File	276
Understanding Accessibility Issues for Websites	278
Meet Chrome’s Live Development Tools.....	280
Troubleshoot CSS with Chrome DevTools	282
Using a Staging Server.....	288
Understanding Front-End Frameworks	290

Index.....	292
------------	-----

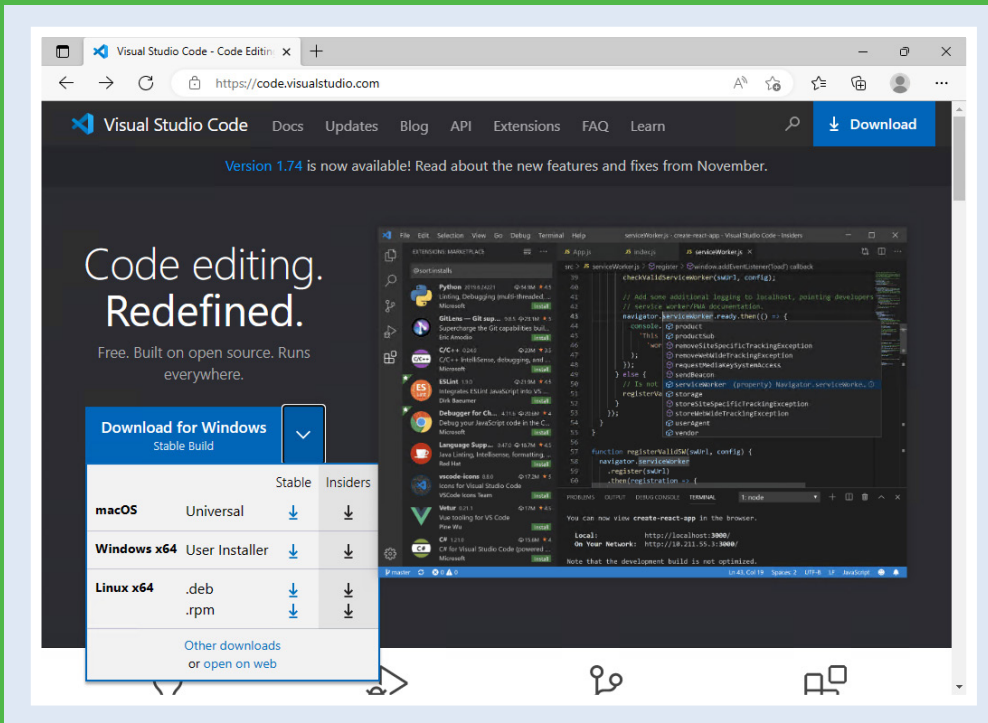




CHAPTER 1

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

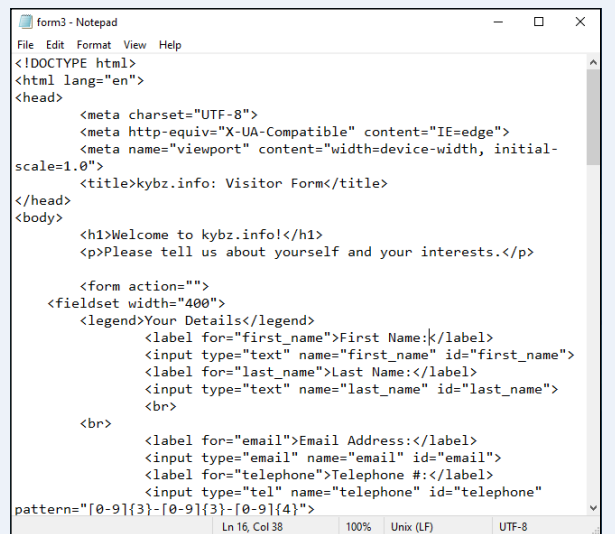
Many different types of tools are available for creating web pages and websites. This section summarizes the various types of tools available and then points you toward the tools this book uses to illustrate creating HTML and CSS files.

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.

A screenshot of the Notepad application window titled 'form3 - Notepad'. The window shows HTML code for a visitor form. The code includes a DOCTYPE declaration, meta tags for charset, compatibility, and viewport, a title 'kybz.info: Visitor Form', and a body with a welcome message and a form. The form has a legend 'Your Details' and three input fields: 'First Name', 'Last Name', and 'Email Address', followed by a 'Telephone #' field. A regular expression pattern is shown at the bottom of the code.

```
<!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>
</html>
```

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.

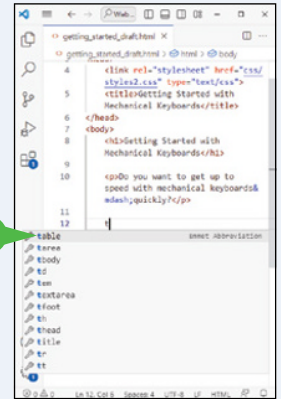
Code Editors

A *code editor* is a text editor enhanced with extra features for creating code — anything from HTML code to programming code. Normally, a code editor is your best choice for creating HTML files and CSS files from scratch and editing them thereafter.

The nearby illustration shows automatic code completion (A) in Microsoft's Visual Studio Code, the free code editor that this book recommends for working with HTML and CSS. See the section "Install Visual Studio Code," later in this chapter, for instructions on putting Visual Studio Code on your computer.

Other widely used code editors include Sublime Text, www.sublimetext.com, which costs \$99 after a free

evaluation without a time limit but with reminders to buy; UltraEdit, www.ultraedit.com, which offers a free 30-day trial and then costs \$79.95 per year for a subscription or \$149.95 for a "perpetual" license; and Notepad++, www.notepad-plus-plus.org, which is free but runs only on Windows.



Website Builders

Website builders are simplified tools that enable you to build a website by dragging and dropping predesigned elements onto a customizable template. Website builders are good for people or small businesses that want to create a straightforward website quickly and without coding.

Widely used website builders include Weebly, www.weebly.com; Wix, www.wix.com; and Squarespace, www.squarespace.com. Most website builders offer web hosting, so you need not find a separate web host.

Many web hosts provide access to one or more website builders. So if you already have a web host, see whether it offers a website builder.

Content-Management Systems

A *content-management system*, abbreviated CMS, is a web-based app for creating, managing, and publishing web pages, blog posts, and images. CMSs provide a wide range of templates for websites, giving you many choices of design and functionality. They also provide tools for

managing and publishing content, including scheduling posts, integrating social media, and SEO optimization.

Widely used CMS platforms include WordPress, www.wordpress.com; Joomla, www.joomla.org; and Drupal, www.drupal.org.

Graphics Tools

To create image files suitable for your website, you will need a graphics-manipulation tool. This book recommends GIMP, the GNU Image Manipulation Program, which is free and runs on Windows, macOS, and Linux. It is available from www.gimp.org; see the section "Install GIMP," later in this chapter.

If you work with graphics professionally, you may already have a suitable graphics-manipulation tool, such as Adobe Photoshop or Adobe Illustrator. Such tools are more than adequate for creating image files for your website. Adobe,

www.adobe.com, offers Photoshop and Illustrator as either single-app subscriptions or as part of a subscription to its Creative Cloud suite of more than 20 apps. Special pricing is available for students, teachers, schools, and universities.

If you have Windows, you might also want to try the built-in Paint app; if it proves inadequate, try the free version of Paint.net from www.getpaint.net. If you have macOS, you might also experiment with the capabilities of the built-in Preview app and the Photos app. For Linux, go straight to GIMP.

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.

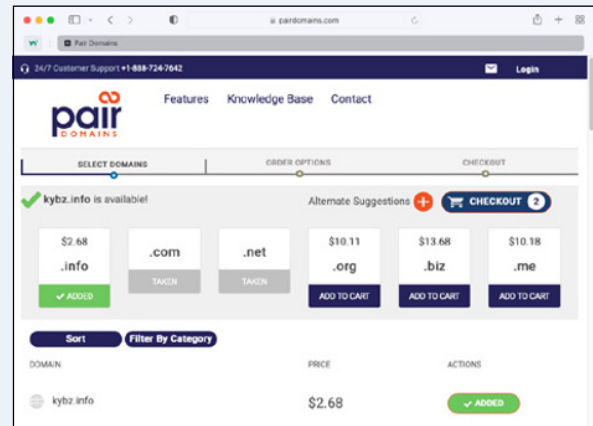
Register a Domain Name

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.

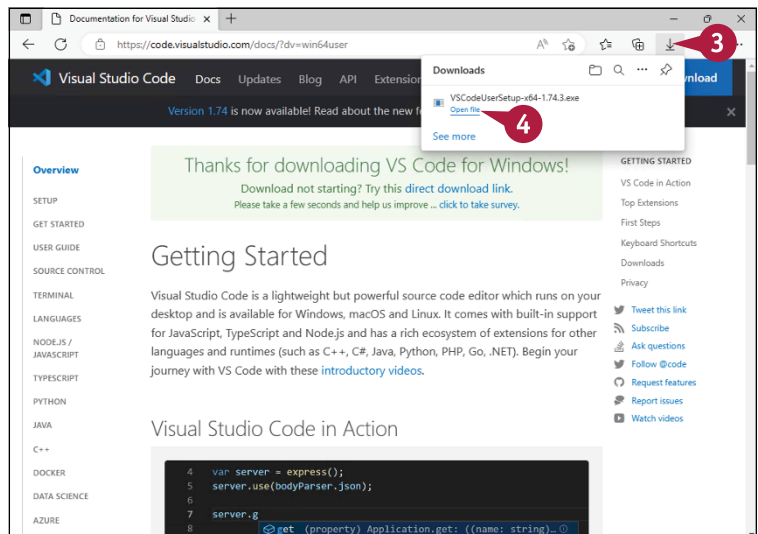
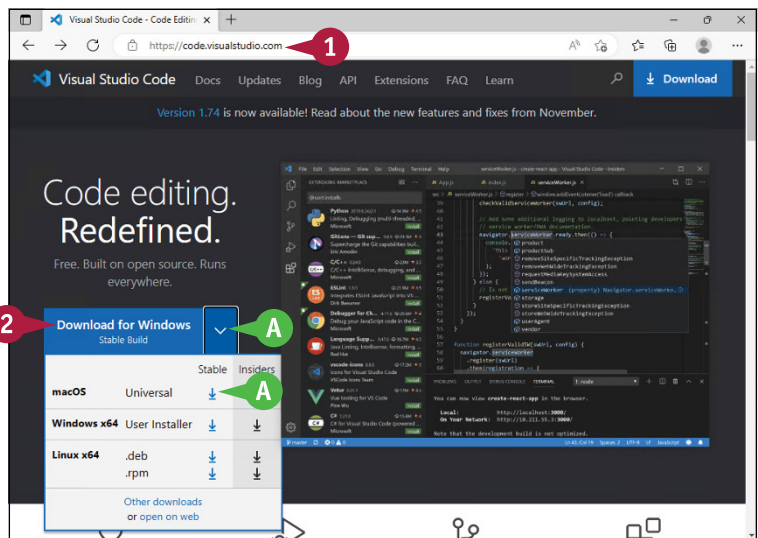
2 Click **Download Stable Build**. This button shows your computer's operating system, which the page automatically detects.

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.

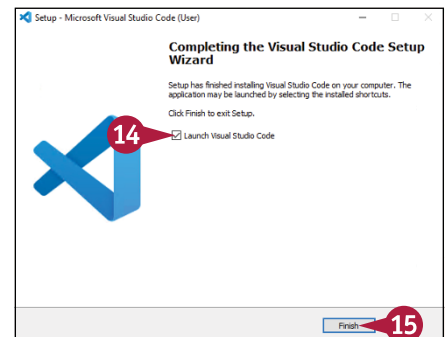
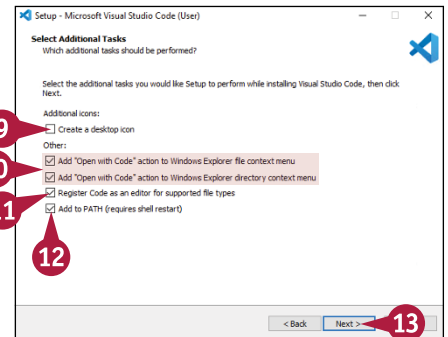
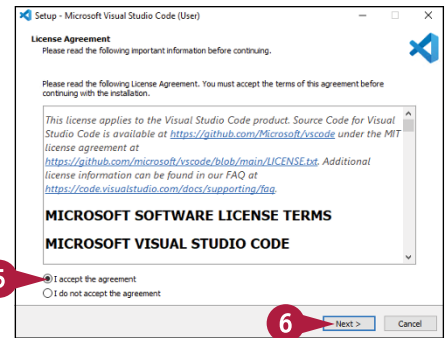
The Completing the Visual Studio Code Setup Wizard screen appears.

- 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

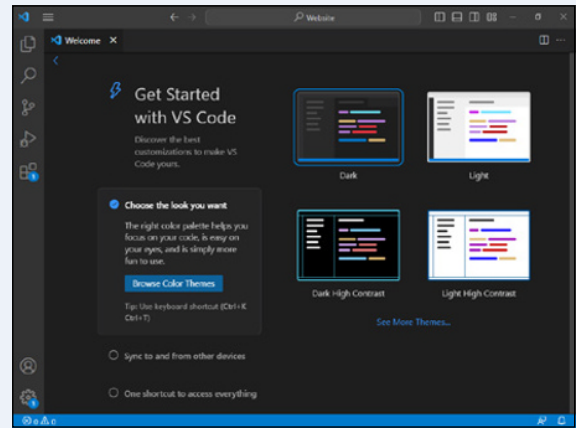
The first time you run Visual Studio Code, the app usually displays the Get Started with Visual Studio Code screen, which walks you through some initial configuration steps. You can return to the Get Started with Visual Studio Code screen later if you like; alternatively, you can use the app's other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

