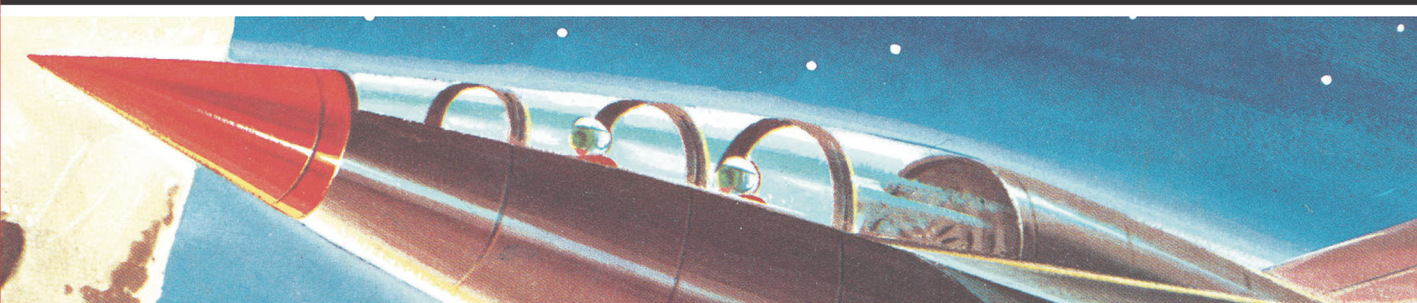


TECH TODAY



PROGRAMMING WITH GITHUB® COPILOT

Write Better Code—Faster!



KURT DOWSWELL

WILEY



Programming with GitHub® Copilot

Write Better Code—Faster!

Kurt Dowsnell

WILEY

Copyright © 2024 by John Wiley & Sons, Inc. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

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

ISBNs: 9781394263370 (paperback), 9781394263394 (ePDF), 9781394263387 (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 and the Wiley logo, 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. GitHub Copilot is a trademark of GitHub, Inc. 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: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993. For product technical support, you can find answers to frequently asked questions or reach us via live chat at <https://support.wiley.com>.

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 Cataloging in Publication data available on request.

Cover image: © CSA-Printstock/Getty Images
Cover design: Wiley



Acknowledgments

Creating this book has been a collaborative effort that would not have been possible without the guidance, support, and encouragement of many people.

First, I want to express my gratitude to Kenyon Brown for believing in my vision and providing me with the opportunity to bring this book to life.

A heartfelt thanks to Satish Gowrishankar for keeping everything on track with meticulous planning, organization, and dedication to deadlines. Your oversight ensured the process remained smooth and effective.

I am deeply grateful to Janet Wehner for coordinating the content and guiding the production through each step with keen attention to detail and precision.

Special appreciation goes to T.J. Corrigan for offering invaluable expertise and meticulously reviewing the technical aspects. Your assistance helped shape this book into what it is today.

To my wife, Paige Lord-Dowswell, thank you for your encouragement, insights, and acumen. You were instrumental in helping me get this book across the finish line.

—Kurt Dowswell



About the Author

Kurt Dowswell is a seasoned software architect with more than 13 years of industry experience. Kurt has spent the larger part of his career leading teams of developers in building, deploying, and maintaining large-scale enterprise software solutions for the U.S. government. He graduated with a BS in computer science from James Madison University.

About the Technical Editor

T.J. Corrigan has worn many hats in his career, including scientist, developer, architect, platform engineer, and engineering lead. Throughout this time, he has always been passionate about improving developer productivity through standardization, automation, self-service, and, most recently, generative AI. T.J. holds a bachelor's degree in computational biology from Carnegie Mellon University. Currently he works as a principal cloud solutions engineer at GitHub, where he helps joint GitHub/Microsoft customers achieve more.



Contents at a Glance

Introduction		xvii
Part I	Getting Started with GitHub Copilot	1
Chapter 1	Get Started with GitHub Copilot	3
Chapter 2	Decoding GitHub Copilot	17
Part II	GitHub Copilot Features in Action	23
Chapter 3	Exploring Code Completions	25
Chapter 4	Chatting with GitHub Copilot	41
Part III	Practical Applications of GitHub Copilot	67
Chapter 5	Learning a New Programming Language	69
Chapter 6	Writing Tests with Copilot	87
Chapter 7	Diagnosing and Resolving Bugs	101
Chapter 8	Code Refactoring with Copilot	113
Chapter 9	Enhancing Code Security	133
Chapter 10	Accelerating DevSecOps Practices	143
Chapter 11	Enhancing Development Environments with Copilot	159
Chapter 12	Universal Conversion with GitHub Copilot	187

Part IV	Key Insights and Advanced Use Cases for GitHub Copilot	215
Chapter 13	Considering Responsible AI with GitHub Copilot	217
Chapter 14	Augmenting the Software Development Life Cycle with GitHub Copilot	229
Chapter 15	Exploring Copilot Business and Enterprise	253
Conclusion		295
Appendix	Resources for Further Learning	297
Glossary		303
Index		311



Contents

Introduction		xvii
Part I	Getting Started with GitHub Copilot	1
Chapter 1	Get Started with GitHub Copilot	3
	Learn Why GitHub Copilot Matters	4
	Create a GitHub Account	4
	Acquire a GitHub Copilot License	4
	Install an IDE Extension	5
	Download Visual Studio Code	5
	Install the GitHub Copilot Extension	6
	Configure the IDE Settings for Copilot	7
	Install Node.js	9
	First Run: Test Copilot	10
	Get the Prerequisites	10
	Explore Copilot	10
	Conclusion	15
	Reference	15
Chapter 2	Decoding GitHub Copilot	17
	Uncover the AI Behind GitHub Copilot	17
	Understand Security, Privacy, and Data Handling	18
	Message Transmission	19
	Data Storage	19
	Prompt and Suggestion Data	19
	User Engagement Data	19
	Additional Security	19
	Vulnerability Prevention System	20
	GitHub Advanced Security	20
	Understand Copyright Protections	20

	Explore the GitHub Copilot Trust Center	21
	Conclusion	22
	References	22
Part II	GitHub Copilot Features in Action	23
Chapter 3	Exploring Code Completions	25
	Introducing Code Completions	25
	Working with Copilot Code Completions	26
	Prerequisites	26
	Naming Your File	26
	Top-Level Comment	26
	Using Meaningful Names	29
	Writing Specific Comments	31
	Referencing Open Tabs	32
	Discovering the Toolbar and Panel	34
	Detailing the Completions Toolbar	34
	Discover the Completions Panel	35
	Updating Copilot Settings	36
	inlineSuggestCount	37
	length	37
	listCount	37
	Leveraging Keyboard Shortcuts	38
	Focus on GitHub Copilot View	39
	Suggest Terminal Command	39
	Triggering Inline Suggestion	39
	Navigate to the Next Panel Suggestion	39
	Navigate to the Previous Panel Suggestion	40
	Open Completions Panel	40
	Conclusion	40
Chapter 4	Chatting with GitHub Copilot	41
	Discovering Copilot Chat	41
	Chatting in the Sidebar	42
	Maximizing Conversations with the Editor View	42
	Expanding Your Conversation to a New Window	43
	Shifting the Conversation to the Right	45
	Utilizing Inline Chat	45
	Discovering Quick Chat	47
	Defining Prompt Engineering with Copilot Chat	48
	Understanding the Basics	48
	Prompting for Code	49
	Prompting for Solutions	49
	Having a Single Clear Objective	49
	Specifying a Specific Technology	50
	Keeping Prompts Short	50
	Gaining Context in Chat	50
	Open Tabs Context	50
	Editor Context	50

File Context	51
Selection Context	52
Commanding Your Conversation with Precision	52
Querying with @workspace	53
Learning with /explain	53
Generating Tests with /tests	56
Finding a Fix with /fix	59
Scaffolding with /new	61
Crafting with /newNotebook	62
Engaging with @vscode	63
Inquiring with /api	64
Learning with @terminal	64
Conclusion	65
Part III	67
Part III Practical Applications of GitHub Copilot	67
Chapter 5	69
Chapter 5 Learning a New Programming Language	69
Introducing Language Education with Copilot	70
Setting Up Your Development Environment	70
Prerequisites	70
Setup Guide	70
Learning the Basics	72
Prerequisites	72
Learning C# with Copilot	73
Creating a Console Application	74
Prerequisites	74
Creating a C# Console Application with Copilot	74
Explaining Code with Copilot	77
Adding New Code	78
Learning to Test	79
Creating Context with Selection	82
Creating Context with Tags	84
Running Tests	84
Conclusion	85
Reference	86
Chapter 6	87
Chapter 6 Writing Tests with Copilot	87
Establishing the Example Project	87
Prerequisites	88
Adding Unit Tests to Existing Code	89
Driving Unit Test Creation with Comments	89
Using Inline Chat to Generate Tests	93
Exploring Behavior-Driven Development with Copilot	94
Adding User Accounts	95
Setup	95
End-to-End Tests	96
Conclusion	99

Chapter 7	Diagnosing and Resolving Bugs	101
	Establishing the Example Project	101
	Prerequisites	102
	Fixing Syntax Errors	103
	Resolving Runtime Exceptions	105
	Setup	105
	Resolving Terminal Errors	109
	Conclusion	111
Chapter 8	Code Refactoring with Copilot	113
	Introducing Code Refactoring with Copilot	113
	Establishing the Example Project	114
	Prerequisites	115
	Refactoring Duplicate Code	116
	Adding Unit Tests	117
	Refactoring Duplicate Error Handling Code	120
	Refactoring Validators	122
	Adding Unit Tests	122
	Extracting Validation Code to Functions	124
	Refactoring Bad Variable Names	127
	Documenting and Commenting Code	129
	Method Documentation	129
	Project Documentation	130
	Conclusion	132
Chapter 9	Enhancing Code Security	133
	Detailing Code Security	133
	Establishing the Example Project	134
	Prerequisites	135
	Exploring Code Security	135
	Using HTTPS	137
	Implementing Validation	138
	Conclusion	138
	Finding and Fixing Security Issues	139
	Fixing Weak Password Hashing	140
	Fixing SQL Injection	141
	Conclusion	142
Chapter 10	Accelerating DevSecOps Practices	143
	Detailing DevSecOps	143
	Simplifying Containers	144
	Creating a Container	145
	Deploying a Container	146
	Applying Security Controls	147
	Automating Infrastructure as Code	148
	Creating IaC	148
	Deploying Code Using Terraform	150
	Applying Security Controls	151

Streamlining CI/CD Pipelines	152
Creating CI Pipeline	153
Adding Security Scanning	154
Creating CD Pipeline	156
Conclusion	158
Chapter 11 Enhancing Development Environments with Copilot	159
Amplifying Visual Studio with Copilot	159
Prerequisites	160
Installing the GitHub Copilot Extension	160
Exploring Code Completions	162
Chatting with Copilot	163
Elevating Azure Data Studio with Copilot	166
Prerequisites	166
Installing the GitHub Copilot Extension	166
Constructing Database Schemas with Copilot	166
Inserting Test Data with Copilot	168
Querying with Copilot	170
Boosting JetBrains IntelliJ IDEA with Copilot	171
Prerequisites	171
Installing the GitHub Copilot Extension	171
Exploring Code Completions	172
Chatting with Copilot	175
Enhancing Neovim with Copilot	176
Prerequisites	177
Installing the GitHub Copilot Extension	177
Exploring Code Completions	177
Consulting Copilot in the GitHub CLI	181
Prerequisites	181
Installing the GitHub Copilot Extension	181
Getting Suggestions with Copilot	181
Explaining Commands with Copilot	183
Setting Up Aliases for Copilot	185
Bash	185
PowerShell	185
Zsh	185
References	185
Conclusion	185
Chapter 12 Universal Conversion with GitHub Copilot	187
Translating Natural Language to Programming Languages	188
Converting JavaScript Components	190
Simplifying CSS Styles	191
Converting Between CSS Frameworks	192
Enhancing Nontyped Languages with Types	196
JavaScript to Typescript	197

	Transitioning Between Frameworks and Libraries	199
	Pandas to Polars	199
	Express.js to Koa.js	201
	Converting Object-Oriented Languages	203
	Objective-C to Swift	203
	Migrating Databases	205
	SQL Server to PostgreSQL	205
	Transitioning CI/CD Platforms	206
	Modernizing Legacy Systems	209
	Modernizing a COBAL Program	209
	Conclusion	213
	Reference	214
Part IV	Key Insights and Advanced Use Cases for GitHub Copilot	215
Chapter 13	Considering Responsible AI with GitHub Copilot	217
	Introducing Responsible AI	217
	Responsible AI Regulation	218
	Examining How Copilot Implements Responsible AI	218
	Fairness	219
	Reliability and Safety	220
	Privacy and Security	221
	Data Protection	221
	Vulnerability Prevention System	222
	Inclusiveness	223
	Transparency	223
	Accountability	223
	Further Research	225
	Programming with AI Responsibly	226
	Researching Responsible AI Principles	226
	Conclusion	226
	References	227
Chapter 14	Augmenting the Software Development Life Cycle with GitHub Copilot	229
	Introducing the SDLC	229
	Requirements	230
	Design	230
	Coding	231
	Testing	231
	Deployment	231
	Maintenance	231
	Assessing the Adoption of AI in the SDLC	231
	Detailing Levels of AI Integration in the SDLC	232
	Level 0: Nonexistent	233
	Capabilities Demonstrated	234
	Level 1: Initial	234
	Capabilities Demonstrated	234

Level 2: Managed	235
Capabilities Demonstrated	235
Level 3: Defined	235
Capabilities Demonstrated	236
Level 4: Quantitatively Managed	236
Capabilities Demonstrated	236
Level 5: Optimizing	237
Capabilities Demonstrated	237
Summary	238
Showcasing GitHub Copilot in the SDLC	238
Detailing the Example Scenario	238
Requirements Gathering	239
Refining the Backlog	242
Planning with Copilot	243
Coding with Copilot	245
Testing with Copilot	246
Deploying with Copilot	248
Addressing Concerns: AI Adoption and the Future of Work	250
Conclusion	251
References	251
Chapter 15 Exploring Copilot Business and Enterprise	253
Introducing Copilot Business and Enterprise	254
Detailing Base Features	254
Programming with Copilot in Your IDE	254
Chatting with Copilot in Your IDE	254
Leveraging Copilot in the CLI	256
Highlighting Copilot Business	256
Presenting Copilot Enterprise	257
Chatting with Copilot in <code>GitHub.com</code>	257
Getting Insights About a Repository	258
Asking Copilot General Programming Questions	260
Indexing Code Repositories to Improve Copilot's	
Understanding	262
Detailing the Example Project	262
Introducing Retrieval-Augmented Generation	263
Indexing Your Repository	264
Asking Repository-Specific Questions	265
How Question	265
Where Question	266
Getting Better Answers with the Knowledge Base	267
Creating a Knowledge Base	267
Summary	273
Leveraging Copilot Chat in Code Repository Files	273
Explaining Code with Copilot	273
Getting Suggested Improvements from Copilot	275
Adding Context with Attach to Current Thread	277

Enhancing Pull Requests with Copilot	279
Discovering Pull Request Tasks	280
Making a Code Change with Copilot	281
Update the Basket Domain Model	281
Update the Basket Service	283
Update the Basket Page	284
Committing to the Feature Branch	285
Leveraging Copilot to Summarize Your Pull Request	286
Managing GitHub Copilot	288
Managing Access	288
Managing Policies	289
Content Exclusion	290
Reviewing Audit Logs	291
Looking Ahead	292
Augmenting Results with Web Search Powered by Bing	292
Customizing Copilot with Fine-Tuned Models	292
Supercharging Copilot with Workspace	292
Conclusion	293
References	293

Conclusion 295

Appendix Resources for Further Learning 297

GitHub Copilot Overview and Subscription Plans	297
Getting Started with GitHub Copilot	298
GitHub Copilot Individual	298
GitHub Copilot Business	298
GitHub Copilot Enterprise	298
Managing Billing for GitHub Copilot	298
GitHub Copilot Product-Specific Terms	298
GitHub Copilot FAQs	299
Community Engagement and Support	299
GitHub Community Discussions on Copilot	299
GitHub Copilot Trust Center	299
Legal and Ethical Considerations	299
Microsoft Copilot Copyright Commitment	299
Empowering Responsible AI Practices	300
Blueprint for an AI Bill of Rights	300
The EU Artificial Intelligence Act	300
Research and Insights	300
What Is Retrieval-Augmented Generation?	300
Quantifying GitHub Copilot’s Impact on Productivity	300
AI’s Impact on the Developer Experience	301
Inside GitHub: Working with LLMs	301
OpenAI Codex	301

Glossary 303

Index 311



Introduction

Welcome to *Programming with GitHub Copilot*, your comprehensive guide to GitHub Copilot. As programming evolves, the tools and techniques at your disposal must adapt to meet the increasing complexity of projects and the demand for faster, more efficient development cycles. GitHub Copilot represents a monumental shift in how code is written, offering you an exceptional partner in your coding journey.

GitHub Copilot is not just a tool; it's transforming the concept of pair programming. Traditionally, pair programming involves two programmers working together at one workstation, continuously collaborating to write better code. However, finding a human partner for this task isn't always feasible. Enter GitHub Copilot, your ever-present AI companion, ready to assist by suggesting code, helping debug, and even writing blocks of code autonomously.

This book is designed to give you practical applications of GitHub Copilot. From setting up your environment to advanced topics like enhancing code security and accelerating DevSecOps practices, each chapter dives deep into real-world applications and provides insights into making the most of this powerful tool.

On the book's website (<https://www.wiley.com/go/programminggithubcopilot>), you will find code files for the starter projects discussed in select chapters. These companion files are designed to help you follow along with the practical examples provided throughout the book.

Whether you're a seasoned developer or just starting out, this book will enable you to harness the capabilities of GitHub Copilot to enhance your programming skills, learn new languages, refactor code, and much more.

Part

I

Getting Started with GitHub Copilot

In This Part

Chapter 1: Get Started with GitHub Copilot

Chapter 2: Decoding GitHub Copilot

Get Started with GitHub Copilot

Software development is full of challenges to overcome. For years, it has been known that programming with a partner can help you learn more, produce better work, and gain more satisfaction while accomplishing your tasks. Although the benefits of pair programming are known, it isn't always possible to have a pair programming partner with you—until now.

GitHub Copilot is your artificial intelligence (AI) pair programming partner, always ready to assist and eager to help you learn! This book will walk you through how to best utilize GitHub Copilot to help you write better code and do it faster.

In this chapter, we will focus on the required steps for getting started with GitHub Copilot. Let's begin!

- Learn Why GitHub Copilot Matters to You
- Create a GitHub Account
- Acquire a GitHub Copilot License
- Install an IDE Extension
- First Run: Test Copilot

Learn Why GitHub Copilot Matters

GitHub Copilot is your AI pair programmer that can assist you in every phase of your software development lifecycle. Whether you are defining your next great feature or configuring a complex continuous integration/continuous delivery (CI/CD) pipeline for an enterprise-grade deployment, GitHub Copilot will be by your side every step of the way, giving you bespoke insights into your business needs. Get ready to take your development productivity and joy of programming to the next level.

You will find your favorite new AI-powered pair programmer, GitHub Copilot, in an ever-growing number of places within your integrated development environment (IDE) and beyond. This book will teach you how to use each Copilot feature in the different license options. We will also explore case studies with best practices that will help to extend your use of Copilot into all areas of your development lifecycle.

To prove the effectiveness of Copilot, a team at GitHub has conducted qualitative and quantitative research to test their hypothesis of improved developer productivity and happiness. One large-scale survey resulted in some amazing results: 88% indicated they were more productive, 74% said they were able to focus on more satisfying work, 96% indicated they were faster with repetitive tasks, and 73% of survey participants indicated they had more time in a flow state [1].

In addition to the survey, the GitHub team conducted a qualitative experiment by having developers create a web server in JavaScript. Individuals using Copilot finished the exercise on average 55% faster [1]! The team used GitHub Classroom to score submissions for correctness and completeness automatically.

Create a GitHub Account

Before you can start using Copilot, you need to have a valid GitHub account. Head to the following web page and ensure that you have access to your account before getting started:

<https://github.com/login>

Acquire a GitHub Copilot License

With a valid GitHub account, we can now review the available licenses for GitHub Copilot. You will need to pick the license that is best for you. There are three GitHub Copilot plans available.

- Copilot Individual
- Copilot Business
- Copilot Enterprise

There are several factors to consider when choosing the correct plan. If you are a student or a maintainer of a popular open-source project, you might be eligible for a free Copilot Individual license.

You can get more information on licenses on this web page:

<https://docs.github.com/enterprise-cloud@latest/billing/managing-billing-for-github-copilot/about-billing-for-github-copilot>

Install an IDE Extension

GitHub Copilot runs as an extension in the following IDEs:

- Azure Data Studio
- JetBrains IDEs (IntelliJ, PyCharm, Rider, and so on)
- Vim/Neovim
- Visual Studio
- Visual Studio Code

NOTE In this book, we will be using the Visual Studio Code IDE for most of the examples. If you use one of the other supported IDEs as your preferred development platform, the information shared in these examples will be transferrable. We will be covering these additional IDEs later in the book when we detail how to set up and configure them to work with GitHub Copilot.

NOTE Support for the JetBrains IDEs is currently in beta.

Download Visual Studio Code

You can download Visual Studio Code (VS Code) from the following page:

<http://code.visualstudio.com>

Once you have installed VS Code on your computer, you should see a welcome screen (see Figure 1.1).

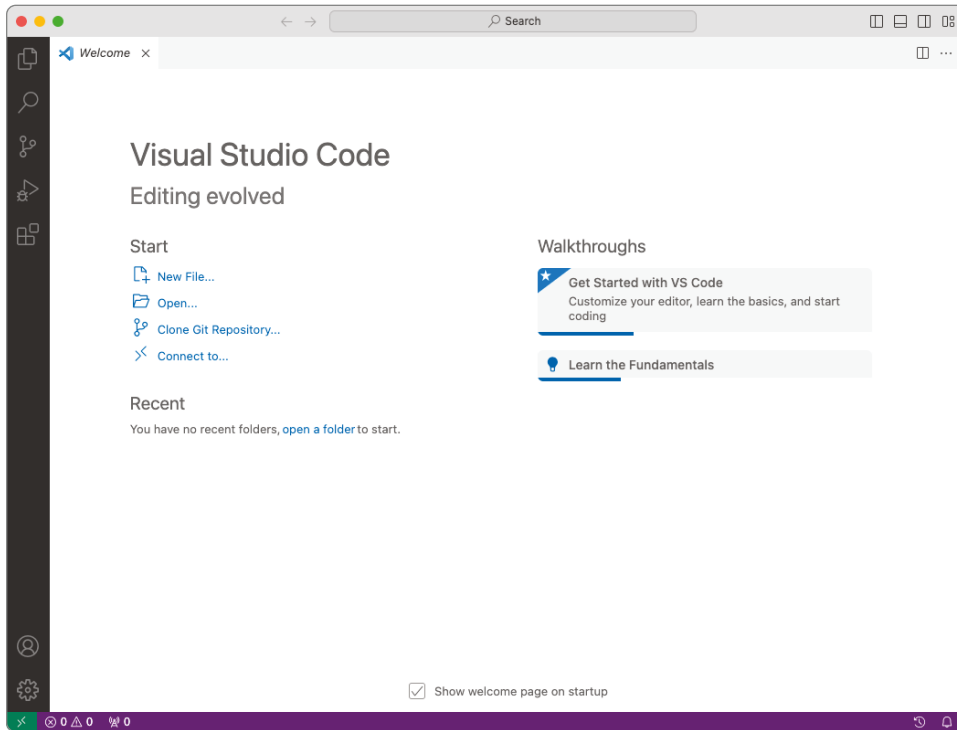


Figure 1.1: VS Code welcome screen

Install the GitHub Copilot Extension

Now that you have the VS Code IDE installed and open, let's navigate to the Extensions panel on the Action Bar. You will find the Extensions panel identified by the "squares" icon.

Now follow these steps:

1. Open the Extensions panel.
2. Search for "GitHub Copilot."
3. Within the GitHub Copilot extension result, click Install (see Figure 1.2).

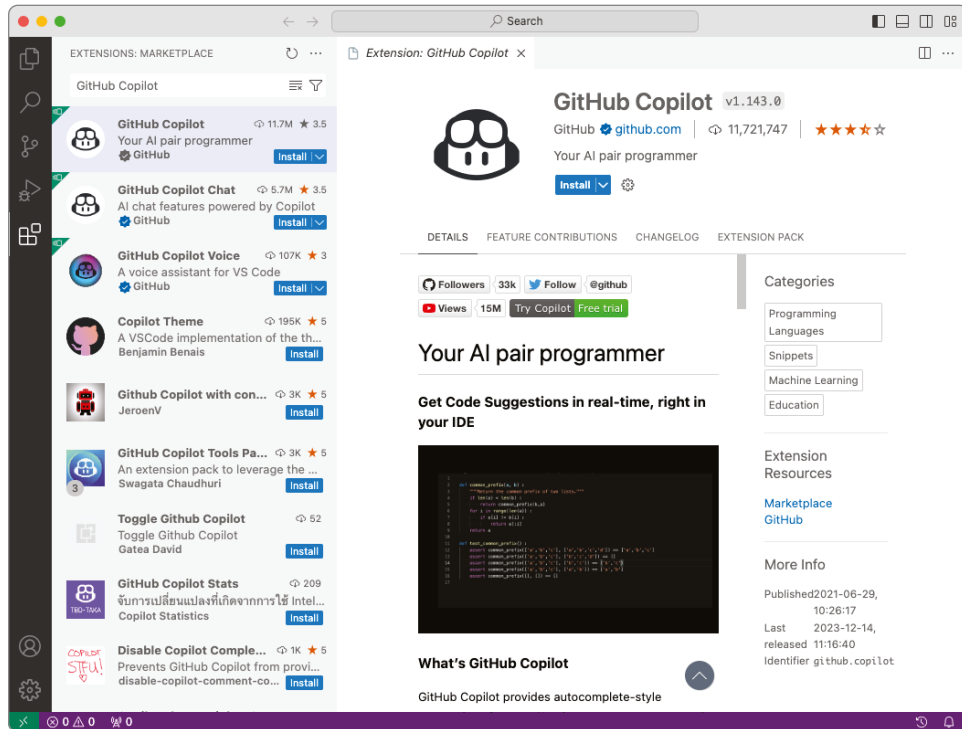


Figure 1.2: VS Code Extensions panel

Configure the IDE Settings for Copilot

After successfully installing the Copilot extension, you are ready to ensure you are authenticated to your GitHub account within VS Code. You should see a pop-up in the lower-right corner of VS Code prompting you to sign in to GitHub (see Figure 1.3). Please use this option to sign in.

If you don't see this prompt after installing the extension, you can also authenticate using the profile menu on the Action Bar (see Figure 1.4).

After completing the sign-in process via the GitHub authentication pages, you can verify your authentication status within VS Code via the bottom-right Copilot icon. Click this icon to bring up the Copilot status menu (see Figure 1.5).

Within the status menu you have access to your status, chat, settings, logs, documentation, and forums.

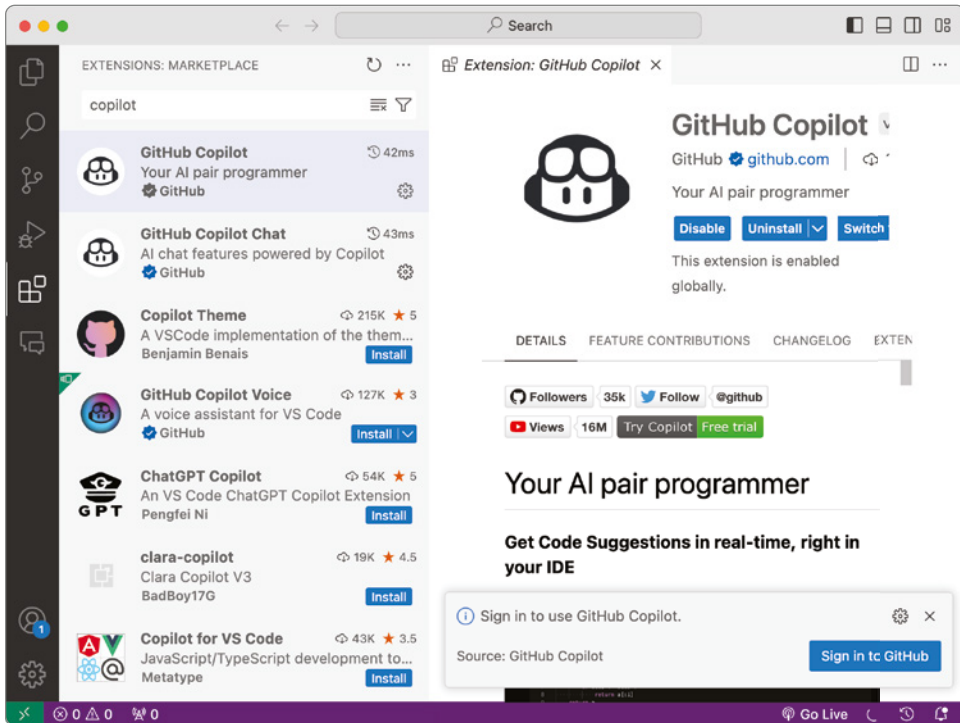


Figure 1.3: Sign-in prompt

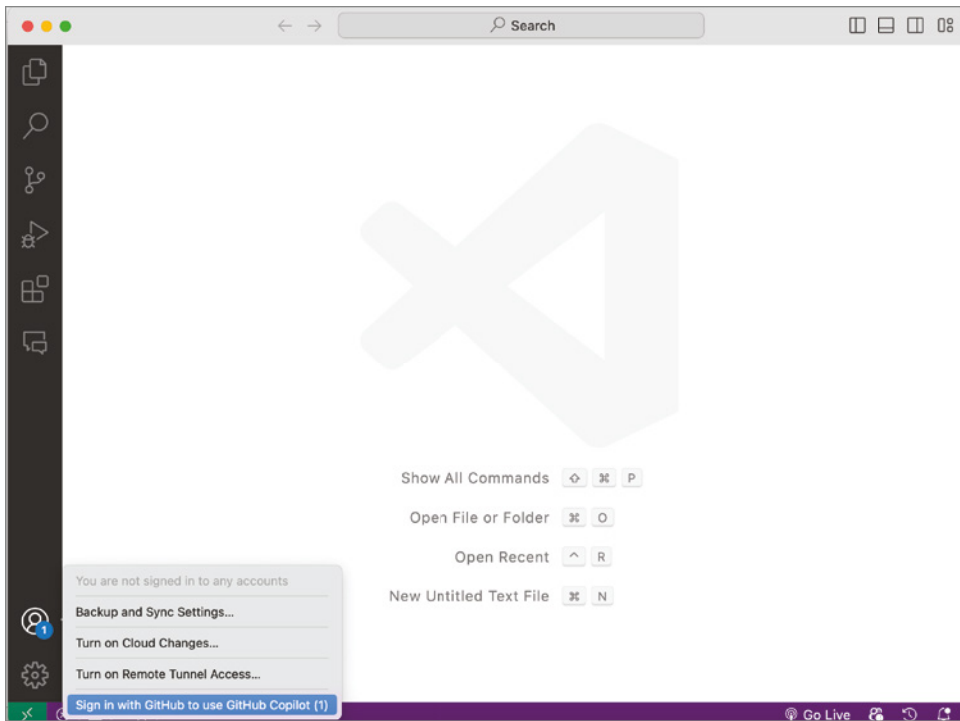


Figure 1.4: Signing in via the Action Bar

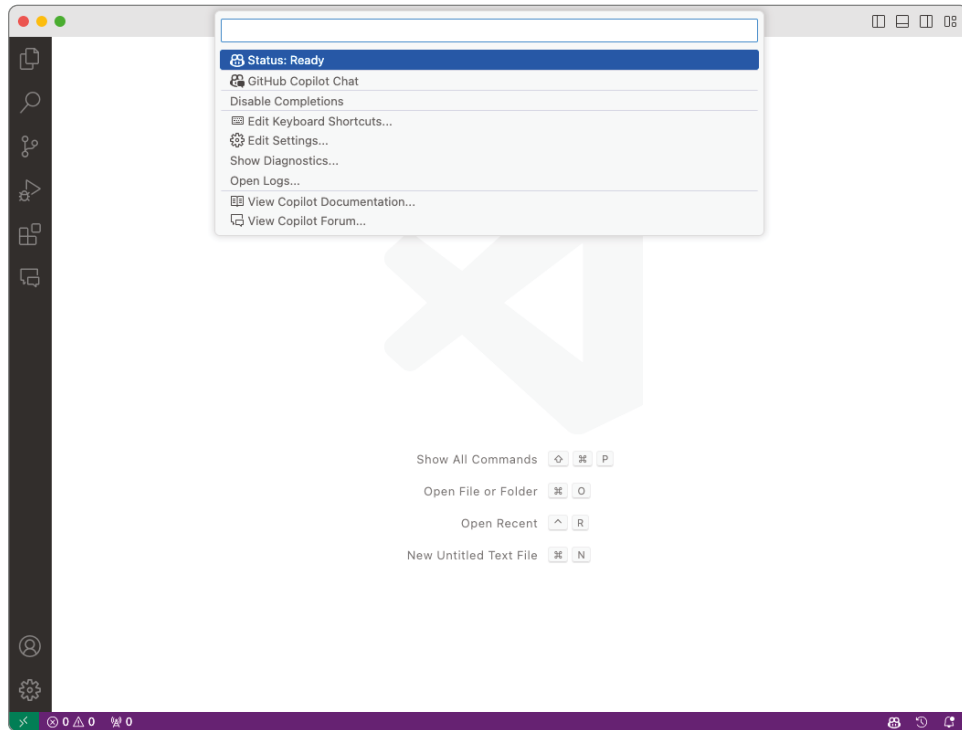


Figure 1.5: Copilot status menu

Install Node.js

Lastly, Node.js will need to be installed to run the example. Node.js is an open-source, cross-platform, back-end JavaScript runtime environment. It allows us to execute JavaScript outside of the web browser.

The easiest way to install Node.js is to go to the website:

<https://nodejs.org/en/download>.

Based on your operating system and computer hardware, select the appropriate download, and follow the installation steps.

After installing Node.js, run the following command in your terminal to confirm that you have installed it successfully.

```
node -v
```

This command will output the node version you have installed.

First Run: Test Copilot

As mentioned, this book will be showcasing the features of GitHub Copilot primarily in Visual Studio Code. There are dedicated chapters later in the book to detail all the other GitHub Copilot IDE experiences.

While most of the code completion features are universal between IDEs, there are differences in the menus, the keyboard shortcuts, and the availability of Copilot Chat (which is available only in Visual Studio and VS Code).

Get the Prerequisites

As mentioned, the following are the prerequisites to testing Copilot:

- VS Code
- GitHub account
- GitHub Copilot license
- GitHub Copilot extension
- Node.js

Explore Copilot

Let's make sure that Copilot is working by writing a quick example function. In this section, you will create a palindrome checker to showcase some of the basic interactions you will have with Copilot within your editor.

Start by opening a folder in VS Code. You can do this via the Explorer menu (see Figure 1.6) or the keyboard shortcut (Cmd+O/Ctrl+O).

NOTE Throughout this book, keyboard shortcuts will be displayed for both macOS and Windows OS.

Create a new folder called **copilot-test** and click Open within your Finder/Explorer window.

Add a new file to your open folder called `palindrome-checker.js` (see Figure 1.7).

Now you are ready to start writing the Node.js script. Let's start by typing a top-level comment in the `palindrome-checker.js` file, as shown here:

```
// node.js application that checks if a string is a palindrome
```