Setting Up GitHub Copilot

Get started with the future of coding by setting up the AI-driven coding assistant, GitHub Copilot!

As a software developer, your job is not just to shape the future of innovation, but also to adapt to the innovations that happen around you. GitHub Copilot is one such innovation—an advanced, AI-powered coding assistant designed to bring your coding experience to new heights.

Key Features and Advantages

As the name suggests, GitHub Copilot is your co-pilot in the world of coding. It's like having a seasoned developer by your side, ready to assist and guide you as you craft your digital creations. With Copilot, you'll have:

- **Code Autocompletion**: As you type, Copilot seamlessly integrates intelligent code suggestions, helping you write code faster and with fewer errors.
- **Code Translation**: Copilot takes natural language comments and translates them into actionable code, bridging the gap between your ideas and implementation.
- **Code Documentation**: In-line code explanations and documentation assistance make understanding and maintaining your code a breeze.

These features allow for many versatile use cases, from expediting code writing, to avoiding or resolving issues in your code. With this available across various programming languages and frameworks, it is also accessible to many developers.

In this article, we'll go deeper into how you can harness the power of GitHub Copilot. Whether you're new to this innovative tool or looking to optimize your existing setup, this guide will walk you through the process, step by step. So, let's get started and make coding a smoother, more efficient experience with GitHub Copilot by your side.

Setting Up GitHub Copilot for VS Code

Now that you have an understanding of what GitHub Copilot brings to the table, it's time to set it up in a code editor! While Copilot can be integrated into most popular editors and development environments, it integrates best

with Visual Studio Code (VS Code), as both were developed by Microsoft. This section will guide you through the necessary steps to get GitHub Copilot up and running seamlessly.

Purchasing GitHub Copilot

Sadly, as the advanced comprehension required by GitHub Copilot needs the resources of many large servers, the service is not free. To purchase a subscription, simply visit the official GitHub Copilot website, sign into your GitHub account, and get started with a free trial. From here you only need to get it running in your development environment.

Installing VS Code

Windows

1. Visit the official Visual Studio Code website.
2. Download the Windows version of VS Code.
3. Run the installer and follow the on-screen instructions.
4. Once installation is complete, launch VS Code.

macOS

1. Visit the official Visual Studio Code website.
2. Download the macOS version of VS Code.
3. Open the downloaded disk image (.dmg) file.
4. Drag and drop the VS Code app into your Applications folder.
5. Launch VS Code from your Applications.

Linux

On Linux, the installation process can vary depending on your distribution. Here's a general guide:

1. Open your terminal.
2. Depending on your Linux distribution, use the appropriate package manager to install VS Code. For example, on Ubuntu, you can use:

```
sudo apt update
sudo apt install code
```

1. Once installed, you can launch VS Code from your applications menu or by running `code` in your terminal.

Accessing Extensions View

With VS Code installed, now is the time to extend its capabilities in the Extensions view. This is where you can manage your VS Code extensions, including GitHub Copilot.

1. Open Visual Studio Code.
2. Click on the square icon on the left sidebar (or use the keyboard shortcut Ctrl + Shift + X on Windows/Linux or Cmd + Shift + X on macOS) to open the Extensions view.

Finding GitHub Copilot

In the Extensions view, you'll find an extensive library of VS Code extensions. To locate and install GitHub Copilot, follow these steps:

1. In the Extensions view, type "GitHub Copilot" in the search bar at the top.
2. GitHub Copilot should appear in the search results. Click on it to view more details.
3. Click the "Install" button and VS Code will download and install GitHub Copilot seamlessly.

Connecting Your GitHub Account

In most cases, GitHub Copilot should prompt you to connect to your GitHub connect upon a successful installation. If this did not happen, follow these steps instead:

1. Open Visual Studio Code if it's not already running.
2. In the top menu, click on "View" and select "Command Palette" or use the keyboard shortcut Ctrl + Shift + P on Windows/Linux or Cmd + Shift + P on macOS.
3. In the Command Palette, type "GitHub: Sign In" and select it from the dropdown list.
4. VS Code will prompt you to sign into your GitHub account. Click on "Sign in with your browser."
5. A web page will open, asking you to log in to your GitHub account if you're not already logged in. Enter your GitHub credentials and click "Sign in."

6. After successful authentication, you'll be asked to grant access to GitHub Copilot for Visual Studio Code. Click on "Authorize Visual Studio Code" to allow the connection.
7. Return to Visual Studio Code, and you should see a confirmation message indicating that you are now signed into GitHub.

Verification and Post-Installation Checks

After the installation process is complete, it's a good idea to verify that GitHub Copilot is successfully installed and ready to assist you with your coding tasks. To do this:

1. Look for the Copilot icon in your VS Code status bar. It should appear as a blue circle with the Copilot logo inside it.
2. Open a code file or create a new one to see Copilot in action. You should notice code suggestions and autocompletions as you type.

If everything works, then congratulations! You've successfully set up GitHub Copilot in your Visual Studio Code environment.

But if instead you are met with confusing errors and issues, try reaching out to the larger GitHub Copilot community, GitHub Support, or referencing the official documentation.

Basic Operations with GitHub Copilot on VS Code

With GitHub Copilot seamlessly integrated into your Visual Studio Code environment and your GitHub account linked, it's time to explore how to use the power of this AI-powered coding assistant in your daily development tasks.

Code Autocompletion

GitHub Copilot excels at code autocompletion, making your coding experience more efficient and error-free. As you type, Copilot analyzes your code context and provides intelligent code suggestions based on appropriate variables, functions, and even entire code blocks. Accept suggestions by pressing Tab or selecting it with your mouse.

Code Translation

One of Copilot's standout capabilities is its ability to transform natural language comments into executable code. To use this feature effectively write descriptive comments by describing what you want to achieve in plain English. Copilot will then translate these comments into functioning code.

Code Documentation

Comprehensive code documentation is crucial for maintaining and collaborating on projects. GitHub Copilot can simplify this process by generating detailed documentation based on comments in the code. Copilot can then further suggest ways to improve code comments and explanations for better code understanding.

Keyboard Shortcuts

GitHub Copilot offers a set of keyboard shortcuts to further enhance your productivity. These shortcuts are specific to Copilot within Visual Studio Code. Some essential shortcuts include:

- **Toggle Copilot Suggestions**: Use Ctrl+Space (Windows/Linux) or Cmd+Space (macOS) to toggle Copilot suggestions on and off.
- **Accept Suggestion**: Press Tab or Enter to accept the currently highlighted suggestion.
- **Show Documentation**: Press Ctrl+ Shift+ H (Windows/Linux) or Cmd+ Shift+ H (macOS) to display documentation for the current code.

Learning from Copilot

GitHub Copilot isn't just a coding assistant; it's also an exceptional learning tool. Consider taking the time to review the code suggestions Copilot provides. This can help you understand coding patterns and best practices. Then use Copilot to experiment with different code suggestions to see how they could benefit your project.

Considerations When Using GitHub Copilot

GitHub Copilot is a powerful tool that can significantly enhance your coding experience, but it's also important to approach its use thoughtfully.

Quality of Suggestions

While Copilot is an exceptional coding assistant, it's essential to remember that the quality of its suggestions can vary. It may not always generate code that aligns perfectly with your project's requirements or coding standards. Security is also a concern as Copilot may suggest code that has security vulnerabilities. Always review the code suggestions provided by Copilot thoroughly. Test the generated code to ensure it functions as expected and adheres to your project's quality standards and follows security best practices.

Data Handling

When using Copilot, be mindful of the data you are sharing with the tool. Copilot operates by learning from publicly available code on the internet, which means it may have access to sensitive or proprietary code. Before committing or sharing any code generated by Copilot, review it carefully to ensure it does not inadvertently expose sensitive information.

Integration with Testing

To make up for some of Copilot's shortcomings, consider using it conjunction with testing frameworks to maintain code quality and security:

- Use Copilot to write unit tests for your code. This ensures that the code functions as intended and helps catch regressions.
- Copilot can assist in generating test cases for various scenarios, aiding in comprehensive testing.
- Monitor code coverage to ensure that all critical code paths are tested thoroughly.

By carefully considering these factors, you can harness the immense power of Copilot while maintaining the integrity and security of your projects.

Wrapping Up

In your journey as a developer, embracing innovative tools like GitHub Copilot can be a game-changer. With its intelligent coding assistance, code

autocompletion, and code translation capabilities, Copilot offers you the opportunity to code more efficiently and effectively.

But Copilot is not just a tool; it's a learning companion. It can help you improve your coding skills over time. By regularly reviewing the code suggestions it provides and experimenting with different approaches, you can enhance your expertise and continue to grow as a developer. Happy coding!