

VS Code for Data and AI Projects

What You'll Learn

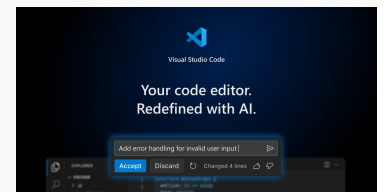
- How to set up VS Code workspace
- How to style and customize VS Code
- How to optimize your workspace
- How to create and work with virtual environments
- How to 2x your productivity with the interactive mode
- How to easily integrate Git into your workflow

Download VS Code

Visual Studio Code - Code Editing. Redefined

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and available on your favorite platform - ...

 <https://code.visualstudio.com>



Introduction

- **Visual Studio Code:** Free integrated development environment (IDE) made by Microsoft for Windows, Linux and macOS.
- **Features:** Supports many languages, debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.
- **Extensible and customizable:** Many settings and huge marketplace with extensions to add new languages, themes, debuggers, and to connect to additional services.
- **Productivity:** Working in VS Code will make you more efficient.

Python Installation

- Ensure you have Python installed on your machine. You can get it through the [official Python installer](#) or by using [Conda](#), a package manager that simplifies installation and management of Python environments.
- The main difference:
 - **Official Python Installer:** Installs Python directly and is best for those who want the pure Python experience or are just starting out.
 - **Conda:** Useful for managing multiple Python environments and dependencies, especially in data science applications. It includes both Python and other tools to get you started quickly.

Command Palette

- The Command Palette in VS Code is a powerful feature that allows you to quickly access commands and features by typing their names. It's like a shortcut hub for all the actions within the editor.
- To open it, press `Ctrl+Shift+P` (or `Cmd+Shift+P` on Mac), and start typing your command. It's awesome because it speeds up your workflow by letting you jump to any function without navigating through menus or remembering keyboard shortcuts.

Workspace Setup

- **Folder Location:** I suggest setting up a `Repositories` folder on your drive to keep all your code projects organized; you can even divide them into Personal Repositories and Work Repositories for better clarity.
- **Opening a Folder:** To open a folder in VS Code using the Explorer, simply navigate to "File" on the menu bar and select "Open Folder," then choose the directory you wish to access.
- **Terminal Shortcut:** Alternatively, you can open a folder directly from the terminal by typing `code .` after installing the command line tools via the Command Palette in VS Code, searching for 'Shell Command: Install 'code' command in PATH', and executing it.
- **Project Template:** For project initialization, I recommend utilizing a GitHub project template. Here's an [example](#).
- **Saving the Workspace:** To save your workspace file along with the folder in Visual Studio Code, first ensure all files are saved, then go to the File menu, select Save Workspace As, choose your desired location, give your workspace a name, and click Save.

Virtual Environments

- **Open Command Palette:** Use the shortcut to open the Command Palette in VS Code.
- **Select Environment Command:** Type and select `Python: Select Interpreter` to choose an existing interpreter or create a new virtual environment.
- **Create New Environment:**
 - **Venv:** Click on Venv → select Python version → requirements.txt (optional)
 - **Conda:** If you have Anaconda installed, it will show up in the list as well. You can select Conda → Python version
- **Check Environment:** After the creation, double check to make sure the right environment is selected. You can adjust using `Python: Select Interpreter`

Installing Extensions

To install an extension in Visual Studio Code (VSCode), follow these steps:

1. **Open VSCode** on your computer.

2. Look for the **Extensions View** icon on the Sidebar, which looks like a square within a square, or press `Ctrl+Shift+X` to open it directly.
3. In the **Extensions View**, you'll find a search bar at the top. Type the name of the extension you're looking for.
4. Browse through the list of extensions that appear based on your search. When you find the one you want, click on it to view more details.
5. Click the **Install** button to add the extension to VSCode.

After installation, some extensions may require additional setup or a restart of VSCode to work properly. You can manage your installed extensions from the Extensions View by right-clicking on them for options such as disable, uninstall, or update.

Recommended Extensions

1. [Python Extension Pack](#)
2. [GitHub Copilot](#)
3. [Path Intellisense](#)
4. [GitHub Pull Request](#)
5. [Better Comments](#)
6. [Ruff](#)
7. [Material Icon Theme](#)
8. [Atom One Dark Theme](#)

Styling VS Code

- **Change Themes:** Settings → Themes
 - Color Theme → Atom One Dark
 - File Icon Theme → Material Icon Theme
- **Customize Icons:** Opens Settings → Cmd/Ctrl + Shift + P → Open User Settings (JSON)
 - Browse the list of options here: [Material Folder Icons](#)

```
"material-icon-theme.folders.associations": {  
  "venv": "environment",  
  "references": "docs",  
  "modeling": "generator"  
},
```

Auto Formatting

- Settings → Format on Save

- Settings → Default Formatter → Ruff

```
"[python]": {  
  "editor.formatOnType": true,  
  "editor.defaultFormatter": "charliermarsh.ruff"  
},
```

Running Jupyter Notebook

- You can run Jupyter Notebooks straight from VS Code and use all the same shortcuts you are used to. Just make sure to select the right kernel.

Interactive Jupyter

- **Interactive Python:** Search for `Jupyter Interactive Window` → Enable (When pressing shift + enter, send selection to Jupyter interactive window as opposed to the Python terminal)
- **Root:** Regarding changing the root folder for Jupyter notebooks, you can modify your settings in VS Code by including the JSON snippet below:

```
"settings": {  
  "jupyter.notebookFileRoot": "${workspaceFolder}/app",  
}
```

GitHub Integration

- **Access Git Commands:** Use the Source Control menu in VS Code to run common Git commands like commit, push, pull, and merge without leaving the editor. This streamlines your workflow by integrating version control into your development environment.
- **Branch Awareness:** Easily check which branch you're working on directly from the Source Control menu. It provides a clear display of your current branch, making it simple to switch branches or create new ones as needed.
- **Open Repo in Browser:** With a quick action from the Source Control menu, you can open your repository in a web browser. This is handy for reviewing pull requests, managing repo settings, or browsing the code on GitHub.