



# An introduction to Visual Studio Code

## Outline

1. [What is Visual Studio Code \(VSCode\)?](#)
2. [Installation](#)
3. [Why should we consider using VSCode?](#)
4. [Purpose of extensions in VSCode](#)
5. [Recommended extensions](#)
6. [Conclusion](#)

## What is Visual Studio Code (VSCode)?

Visual Studio Code (VSCode) is a free, open-source code editor developed by Microsoft. It's lightweight, highly customisable, and supports a wide range of programming languages and file types.

### Installation:

#### Windows:

1. Visit the official [VSCode website](#).
2. Download the installer for Windows.
3. Run the installer and follow the on-screen instructions.
4. Once installed, launch VSCode.

#### macOS:

1. Visit the official [VSCode website](#).
2. Download the installer for macOS.
3. Open the downloaded .dmg file, and drag the VSCode icon to the Applications folder.
4. Launch VSCode from the Applications folder.

## Why should we consider using VSCode?

1. Editing with ease: VSCode provides a powerful code editor, offering features like *IntelliSense*, debugging, and an interactive console. While we'll predominantly be using it for Python, it supports multiple other programming languages, such as SQL, Java, and R.
2. Jupyter Notebooks support: VSCode supports Jupyter Notebooks, allowing us to create, edit, and run notebooks in the VSCode environment. This is especially useful when an advanced text editor is required.
3. Python support: VSCode has robust support for Python, making it easy to write, test, and debug Python code.
4. Extensions: Apart from the specific reasons listed above, VSCode has a multitude of extensions available to improve your coding experience.

## Purpose of extensions in VSCode

Extensions serve to enhance and customise the editor's functionality. They are add-ons that can be installed to tailor VSCode to our specific needs. Here are instances where we'd make use of extensions:

1. Language support: Extensions provide language support for a wide range of programming languages, including syntax highlighting, autocompletion, and documentation lookup.
2. Integrated tools: Extensions integrate external tools (such as GitHub for version control) directly into VSCode, enhancing its capabilities for specific tasks like linting, formatting, and testing.
3. Customisation: Extensions allow us to customise the appearance and behaviour of VSCode. Themes, icon sets, and keymap configurations can be easily modified.
4. Specialised workflows: There are extensions tailored for specific workflows and frameworks. For example, extensions for web development, machine learning, or Docker integration.
5. Enhanced productivity: Many extensions focus on improving developer productivity by adding features such as snippet support, intelligent code navigation, and quick documentation lookup.

## Recommended extensions:

Although there are a multitude of extensions available for various languages, functions, and integrations in VSCode, we're only mentioning three to get you started:

### Python extension for Visual Studio Code:

Provides enhanced support for Python, including linting, IntelliSense, debugging, and Jupyter Notebooks integration. Install by searching for "Python" in the Extensions view (Ctrl+Shift+X).

### Jupyter:

Enhances Jupyter Notebook support within VSCode. Search for "Jupyter" in the Extensions view.

### GitLens - "Git supercharged":

Provides powerful Git capabilities to VSCode, enhancing version control features. Search for "GitLens" in the Extensions view.

## Conclusion:

Visual Studio Code is a simple but flexible code editor that provides a fantastic development environment for Python, Jupyter Notebooks, and almost all other coding languages. Its ease of use, extensibility, and active community make it a popular choice among developers. Version control is simplified with extensions, and it's easy to create a development environment tailored to a specific project's needs. Give it a try!

