

The VS Code Text Editor

We'll be writing Python scripts using software called a ["text editor"](#). Similar to word processing software like Microsoft Word, text editors allow us to write and save documents of text. But unlike word processors, which save extra metadata (e.g. styles and formatting) along with the underlying text, text editors allow us to save files comprised of just text. This helps the computer more easily interpret them.

There are many good text editor options out there, including [VS Code](#), [Atom](#), [Sublime](#), and [Notepad++](#), and it seems each developer has their own preference.

Regardless of which text editor you choose, you are highly encouraged to configure it with certain plugins, packages and extensions to enhance your experience and save you time. There are two sets of functionality in particular which will prove especially valuable this semester. These are "Python syntax auto-completion" and "column-style selection", respectively (see basic configuration section below).

Installation

[Download VS Code](#).

Basic Configuration

After you have downloaded VS Code, you'll want to take some time to familiarize yourself with its [settings](#) and menus.

The Command Palette (accessible by typing "shift + command + P") is perhaps the biggest time-saving tool, and is worth exploring.

Shell Commands

On a Windows, you may not need to take any action. But on a Mac, follow [these steps](#) to enable VS Code shell commands.

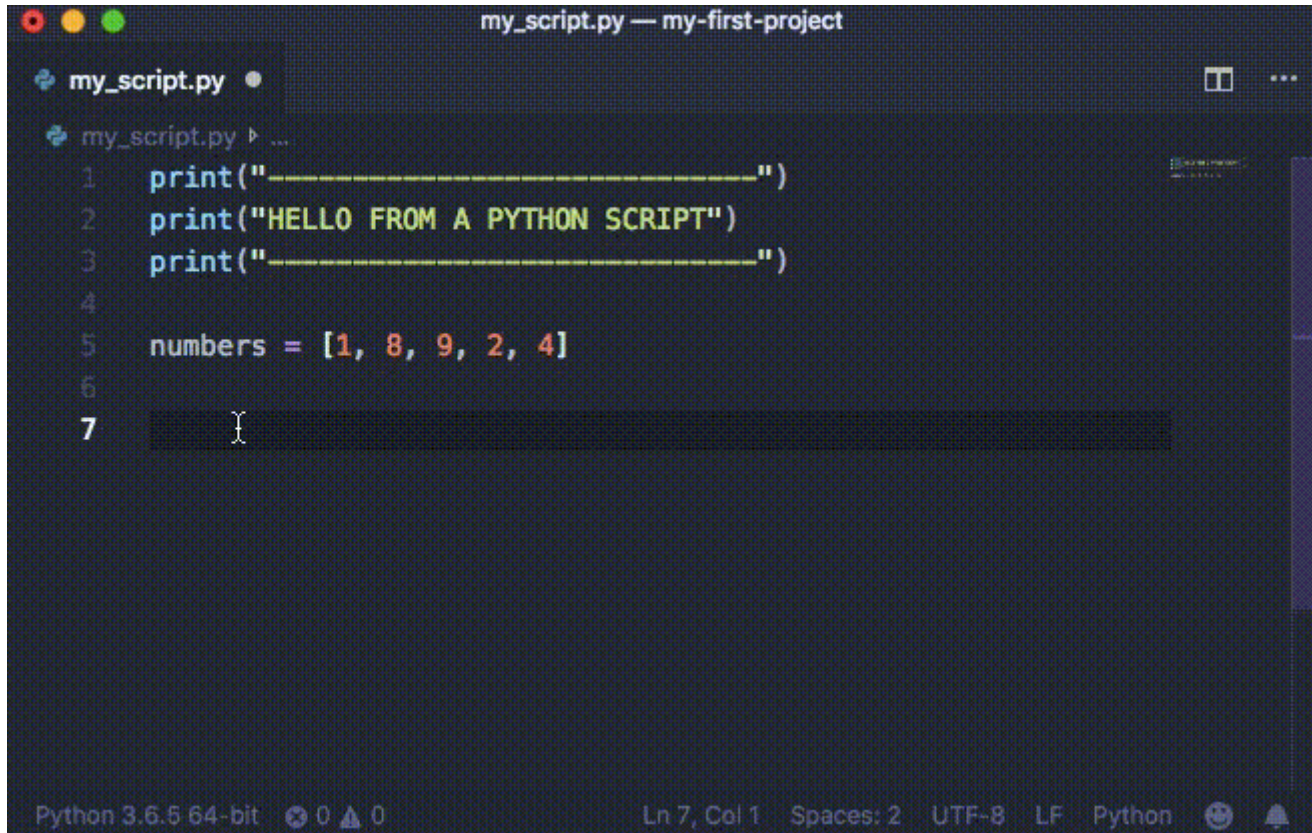
After enabling shell commands, you should be able to use the [code](#) command to open files and folders from the command-line:

```
# open all files and folders in the current working directory:
code .

# open all files and folders in the specified directory, e.g. "path/to/my-project":
code path/to/my-project
```

Python Syntax Auto-completion

Once configured, the text editor is capable of automatically completing snippets of Python code for you. This helps improve accuracy, and saves time.

A screenshot of the Visual Studio Code editor interface. The title bar at the top says "my_script.py — my-first-project". The editor window shows a file named "my_script.py" with the following code:

```
1 print("-----")
2 print("HELLO FROM A PYTHON SCRIPT")
3 print("-----")
4
5 numbers = [1, 8, 9, 2, 4]
6
7
```

The cursor is at the end of line 7. The status bar at the bottom indicates "Python 3.6.5 64-bit", "0 errors, 0 warnings", "Ln 7, Col 1", "Spaces: 2", "UTF-8", "LF", and "Python".

When you open a Python file in VS Code, it should prompt you to install the official Python extension ([ms-python.python](#)). You are recommended to install this extension to enable Python syntax auto-completion.

Column Selection

If configured, your text editor can also enable vertical text selection. This comes in handy if you have to change multiple lines of text at the same time, including commenting-out many lines at once.



```
my_script.py — my-first-project
my_script.py x
my_script.py ▶ ...
1 print("-----")
2 print("HELLO FROM A PYTHON SCRIPT")
3 print("-----")
4
5 teams = [
6     {"city": "New York" , "name": "Yankees" },
7     {"city": "New York" , "name": "Mets" },
8     {"city": "Boston" , "name": "Red Sox" },
9     {"city": "Washington" , "name": "Nationals" },
10    {"city": "New Haven" , "name": "Ravens" }
11 ]
12
13 print(teams)
14
Python 3.6.5 64-bit 0 0 1 Ln 11, Col 2 Spaces: 2 UTF-8 LF Python
```

By default, you should be able to achieve column selection functionality in VS Code by pressing "shift + alt" then clicking and dragging up or down.

Further Optional Configurations

These are some of the professor's personal VS Code configurations, for your reference. Feel free but not obligated to use them.

Extensions

A sample of the professor's installed extensions (results from running `code --list-extensions`):

```
HookyQR.beautify
mikestead.dotenv
ms-python.python
ms-vscode.atom-keybindings
sleistner.vscode-fileutils
streetsidesoftware.code-spell-checker
whizkydee.material-palenight-theme
xamm.filepath
yzhang.markdown-all-in-one
```

You might try searching these manually or [importing them programmatically](#) via `code --install-extension EXTENSION_NAME`, where "EXTENSION_NAME" is the extension's identifier (see list above).

DISCLAIMER: the professor has not tested the extension importing process, so if you experience issues, please report them immediately

User Settings

A sample of relevant settings from the professor's `settings.json` file:

```
{
  "workbench.activityBar.visible": false,
  "workbench.colorTheme": "Palenight Theme",
  "breadcrumbs.enabled": true,
  "editor.fontSize": 14,
  "editor.wrappingIndent": "none",
  "editor.formatOnPaste": true,
  "editor.formatOnType": true,
  "editor.tabSize": 4,
  "explorer.confirmDragAndDrop": false,
  "explorer.confirmDelete": false,
  "files.trimTrailingWhitespace": true,
  "files.insertFinalNewline": true,
  "files.trimFinalNewlines": true,
  "emmet.showSuggestionsAsSnippets": true,
  "diffEditor.ignoreTrimWhitespace": false,
  "window.zoomLevel": 0,
}
```