

PHY1112 Lab 1

> Hello World!

January 9th, 2024

Part	1	2	3	4	Total
Points	1	1	1	2	5
Score					

Objectives

1. Assure the existence of a Python installation on a system.
2. Configure and use VSCode to work with and create Python files.
3. Write a Python file which can output any general string to the console.
4. Execute the file using both command line instructions and the VSCode Python interpreter.

Part 1: Assuring Python exists

Whenever first embarking on writing Python code on a new machine, it is important to assure that Python exists, and is of a version that will work for your goals. Depending on the operating system (OS) of the computer, there are a few ways this can be done. The computers in the lab are all running Windows 10, so here are directions for the Windows family of OSes¹:

First **open a command prompt terminal** from the start menu by searching “cmd”.

To assure Python is installed, **type the following command** (exclude the “\$”, it is only there to indicate that this is a command which is typed on the command line without administrator or superuser privileges):

```
$ python.exe --version
```

If Python is installed, this will **print the version number for Python**.

1. (1 point) What is the version of Python installed on the machine?

¹ If you want to confirm Python's existence on your computer at home, and you have a non-Windows family OS, see the “Appendix: Command alternatives on OSes other than Windows” section at the end of this document.

Part 2: Configuring VSCode

VSCode is opened by **searching for “code” in the start menu search menu**. It should be installed on the system and opening it will bring you to the welcome screen. Since we will be working with Python code, we will want to install the Python extension.



There will be a vertical tab bar on the left side of the screen (see figure at left). **Click on the extensions tab** (see yellow box in figure at left for icon location)²

In the search bar at the top of the side window, **type the phrase “python” and hit enter**.

A list of extensions will show up. **Click the one titled “Python” which is authored by “Microsoft”**.³

Once in the extension tab, **click the “Install” button**. If there are only “Disable” and “Uninstall” buttons, then the extension is already installed and nothing more needs to be done. To close out of the extensions tab, click on the X next to the tab title.

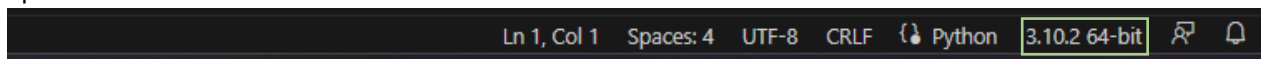
In the menu at the top of the screen, **click File→Open Folder**. . . and navigate to your user directory.

Create a new folder there titled “Lab1” and select that folder.

On the vertical tab bar, **click on the explorer tab** (see the red box at left)⁴.

Create a new file by right-clicking in the empty area and selecting New File...⁵. **Title the file “lab1.py”**, and the file will open automatically in the main part of the window.

At the bottom of the screen there will be an info bar that displays information relating to the currently opened file:



There should be a number on the right (see green box above) that should correspond to the Python version installed. If not, there will be text that reads “Select an interpreter...” In either case, **click the text**⁶

This brings up a prompt that allows you to select which Python interpreter you want to use for this Python file. There may be multiple options or there may only be one option. **Select the option corresponding to the version of Python you reported in question 1**. You will see a greyed-out path next to the interpreter name that shows the location of the interpreter you are selecting.

2. (1 point) What is the path to your selected Python interpreter?

² Or use the keyboard command Ctrl+Shift+X

³ Notice the checkmark symbol next to the author name. This checkmark indicates that the extension is maintained by a verified author, and extensions with this checkmark are generally trustworthy.

⁴ Or use the keyboard command Ctrl+Shift+E.

⁵ Or use the keyboard command Alt+Ctrl+N

⁶ Or use the keyboard command Ctrl+Shift+P and type “Python: Select Interpreter” and hit enter.

Part 3: Writing the Python script

Now that you are in the Python file and your coding environment is setup, the next step will be to write code that will print a string to the console. You can choose any string, although the traditional first string to print is simply "Hello World!".

While this is an easy task, it is not always intuitive to remember syntax for every command. If you want assistance on base Python syntax, go to <https://docs.python.org/>, and use the quick search at the top to search for the function you want to use.

Alternatively, you can view all the built-in functions to the Python standard library by going to <https://docs.python.org/3/library/functions.html>. Make sure that the appropriate Python version is selected from the drop down in the upper left corner. This website also provides tutorials, how-to's, in-depth setup directions, etc., and is generally a good resource for new Python users.

3. (1 point) Which of the following is valid syntax for a one-line print statement which prints > hello in Python 3.11+? [Note that the ">" character represents that what follows is a command line output, and is not part of the output itself.]
- A) `print(hello)`
 - B) `print("hello")`
 - C) `print "hello"`
 - D) `print hello`

Part 4: Executing Python code

The goal of this part will be to **execute the print statement you chose for part 3**.

There are multiple ways to execute code in a Python file. The most common way to execute code directly from a Python file is via the **command line**. While the software where the commands are executed differs between operating systems, the general commands themselves are nearly identical.

Windows: the command prompt (cmd) is typically the chosen command line interface, however the Powershell is becoming more popular with recent OS versions (Windows 10+).

MacOS and Linux: the command line interface is instead referred to as a terminal emulator, or more simply just a terminal. On **MacOS** machines, the stock Terminal app works as a very effective terminal emulator, though others exist that can be installed. On **Linux** based operating systems, there are a wide range of different terminal emulators, with xterm, gnome-terminal, and rxvt-unicode being popular choices.

Regardless of the terminal emulator or command line interface that is used, VSCode is able to interface with them directly. To open the terminal within VSCode, navigate to the top menu and **select**

Terminal→New Terminal⁷

When the terminal opens, you will notice that it will open to the current directory that you created earlier, where your “lab1.py” file is located⁸. **To confirm your file is in this directory, use the following command** on Windows⁹

```
$ dir
```

that will list all the files in the current directory. **To execute your code, use the following command**

```
$ python.exe lab1.py
```

after which, either the output of your print statement will be displayed, or an error will be displayed.

Alternatively, and more simply, you can execute your Python file with one click or keyboard command using VSCode. In the upper right corner of the Python file tab, there is a play button with a drop down arrow next to it. **Clicking the play button** directly will create a new terminal (or use an existing one) and execute the Python file¹⁰

4. (2 points) Show the TA the results of executing the code that prints “Hello World” using both methods.

Appendix: Command alternatives on OSes other than Windows

In MacOS and Linux based systems, there is slightly different syntax when it comes to using commands in the terminal. For the Python commands, you can simply type the same commands without the “.exe” extension, for example

```
$ python -version
$ python lab1.py
```

In fact, even on most modern Windows systems this can generally be done as well, however including the “.exe” extension is compatible with all Windows systems when using cmd. When using the new Powershell in Windows, the Python commands instead become

```
$ py -version
$ py lab1.py
```

When attempting to list the contents of a directory on either MacOS or Linux based systems, as well as when using the Windows Powershell, the following command should instead be used

```
$ ls
```

⁷ Or use the keyboard command Ctrl+` (the dash in this keyboard command is called a grave and is generally on the same key as the symbol ~).

⁸ If you need to change directories, the `cd` command can be used to do so (details of its use can be found by searching online).

⁹ see “Appendix: Command alternatives on OSes other than Windows section” for other operating systems.

¹⁰ Or use the keyboard command Shift+Enter.