

CSE 20

Beginning Programming in Python

Lab Assignment 1

The purpose of this assignment is to learn how to run Python in both interactive mode and script mode, and to submit a Python program for this class. Begin by going to the Python website <https://www.python.org/>. Download and install Python 3.8.5 on your computing device. (If you're using Windows, be sure to check the box "Add Python 3.8 to PATH" during the installation process).

A *command line environment* or *command interpreter* is an application into which you can type commands to your operating system. The basic command interpreter for Mac is called *terminal*. Windows systems have a number of different command line environments. Here we illustrate the Windows *powershell* command interpreter. The main difference between the two is that terminal in Mac is a version of the *Unix* operating system, while powershell in Windows is not. Each command interpreter has its own language, but many of the most basic commands work in both systems. Search for and open the command line tool for your system (terminal for Mac, or powershell for Windows).

Once your command line environment is open you will see a *command prompt*, which may be "\$", or ">", or some other (sequence of) symbol(s). This prompt is just the operating system saying it's ready for you to type a command. In all examples, you **do not** type this symbol. At the command prompt, type `python` (in powershell) or `python3` (in terminal) to start Python in *interactive mode*.

```
> python
or
$ python3
```

You will now see a new command line environment called the *Python Interpreter* or *Python Shell*. Its command prompt is ">>>" (on all systems). Upon first starting Python, you will also see which version of Python is running. For example

powershell:

```
> python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

terminal:

```
$ python3
Python 3.8.2 (default, Jul 16 2020, 14:00:26)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Your actual results may differ. Pay attention to the version number though, which should be 3.8.x. If you installed the most recent version, that last x will be Linux 5, but any value of x is fine. The main thing is to make sure you are not running any version of Python 2.

Emulate everything I did in interactive mode on days one and two of the quarter. If you don't remember exactly what I did, see the lecture recordings by going to YuJa on the webpage, or in Canvas. Use what you learned in lecture to calculate the value of the arithmetic expression: $3^{7!} - 2^{6!}$. Take a screenshot of your Terminal that includes both the expression you typed and the answer, and save it as an image file on your computer. If you don't know how to take a screenshot, see

<https://www.theverge.com/2019/11/8/20953522/how-to-take-screenshot-mac-windows-pc-iphone-android>

This completes the interactive mode portion of the assignment. You can quit the Python shell by typing

```
>>> exit()
```

At this point you are back to your terminal (\$) or powershell (>) command interpreter.

A Python program is a text file containing a sequence of Python commands. Such a file must always end with the .py suffix. Running Python in *script mode* simply means running such a program. Go to the Examples link on the class webpage

<https://classes.soe.ucsc.edu/cse020/Fall20/Examples/>

and save the file HelloWorld.py to your computer. Do not copy-paste the contents of this file from your browser into an editor. It's much better to save the file directly (using `ctrl-s` in most browsers.) For convenience, save the file to your Desktop. To run this program you must first navigate your command interpreter into your Desktop folder.

Before we can do that, recall there are a number of basic commands that are recognized by both terminal and powershell. Among these are

<code>pwd</code>	(print working directory, i.e. where am I)
<code>ls</code>	(list the contents of the current directory, i.e. what is here)
<code>cd</code>	(change directory, i.e. move someplace else)
<code>man</code>	(display manual pages for a command)

Type `pwd` at your command prompt, for instance, and observe the results. Also try `ls` and `cd`, respectively. Now type `man pwd`, `man ls`, `man cd` and `man man` (on separate lines of course) to get a feel for what man pages look like. This form of documentation may appear impenetrable to beginners, since they seem to assume a great deal of knowledge on the part of the user. Don't worry about this. We may tackle man pages at some later time, but not now. Recall that a command interpreter is actually a full and complete computer language. These four commands (`pwd`, `ls`, `cd` and `man`) are some of the most common Unix commands. If you are a Windows user, you may be wondering how these Unix commands can be part of the powershell language. Actually the equivalent powershell commands have different names, but the Unix versions are recognized as a convenience.

Unix	PowerShell	Meaning
<code>pwd</code>	<code>Get-Location</code>	display name of current directory
<code>ls</code>	<code>Get-ChildItem</code>	display contents of current directory
<code>cd</code>	<code>Set-Location</code>	move to another directory
<code>man</code>	<code>Get-Help</code>	display man pages

This is all I'll say about the native powershell commands, since they are only relevant to Windows users. I will continue to use the Unix equivalents in both powershell and terminal.

Now let's focus on the `cd` command. Notice that if you do `pwd` then `cd` then `pwd`, it appears that `cd` doesn't do anything. That's because `cd` needs a command line argument telling it where to move to. Do the following at your command prompt.

```
$ pwd
$ cd ~/Desktop
$ pwd
$ ls
```

(Remember do not type the command prompt `$`.) The result for Mac should resemble

```
$ pwd
/home/ptantalo
$ cd ~/Desktop
$ pwd
/home/ptantalo/Desktop
$ ls
HelloWorld.py
$
```

As usual, your results may be quite different, depending on what other files you have on your Desktop (and will look especially different in Windows.) Of course `ptantalo` will be replaced by your user name on your own system. Observe that `~` stands for your home directory, so after the second `pwd`, you are located in the directory `Desktop` contained in your home directory.

You are now ready to run the Python program `HelloWorld.py`. Do `python HelloWorld.py` in Windows powershell, and `python3 HelloWorld.py` in Mac terminal. The results should be the same in both systems.

```
> python HelloWorld.py
Hello, World!
>
```

or

```
$ python3 HelloWorld.py
Hello, World!
$
```

You now know how to run a Python program from your command line environment. Our next goal is to edit this program so that it prints out your email address, instead of the message `"Hello, World!"`. To do this you will need a *Code Editor*. There are many such applications available. I recommend

Notepad++	Windows only
BBedit	Mac only
SublimeText	all platforms

There are many others. Also, any good IDE (Integrated Development Environment) like PyCharm or Spyder has a code editor built in. Download and install an appropriate code editor and open `HelloWorld.py` in it. Change its text from

```
#-----
# HelloWorld.py
#-----

print("Hello, World!")
```

to

```
#-----  
# Email.py  
#-----  
  
print("CruzID@ucsc.edu")
```

Where `CruzID@ucsc.edu` is your actual UCSC email address. Save the file under the new name `Email.py`. You should now have a new file on your Desktop with that name. Run this new program in your command line environment as before. Your result should be similar to

```
> python Email.py  
ptantalo@soe.ucsc.edu  
>
```

or

```
$ python3 Email.py  
ptantalo@soe.ucsc.edu  
$
```

Of course instead of my email address, it should print out yours. If you get any kind of error, check your typing carefully. Notice that all we did was change the comment block so that it states the new file name, and change the string inside quotes in the `print()` statement.

Rename your screenshot image file to `lab1-image.xxx` (where the file name suffix `.xxx` matches the kind of image file it is, like `.jpg`, or `.png`). Now, in your web browser, go to

Canvas -> Gradescope -> Assignments -> lab1

and upload the two files `lab1-image.xxx` and `Email.py` as attachments to questions 1 and 2, respectively. Do all of this before the assignment due date (stated on the webpage.) That's it, you're done. You can now (1) run Python in interactive mode, (2) run a Python program from within your command line environment and (3) submit an assignment to this class.