

Python on Windows 10

Grant Esparza

gesparza3@mail.csuchico.edu

April 4, 2018

Getting Ready

In order to install the bash shell command-line tool, you must have the **Windows 10 Creators Update** installed. In order to check if your PC meets the requirements:

1. Open **Settings**.
2. Click on **System**.
3. Click on **About**.

Under version, you must have at least version **1703**. If not, you must update your PC.

Installing Bash command-line tool

1. Open **Settings**.
2. Click on **Update & security**.
3. Click on **For Developers**.
4. Under "Use developer features", select the Developer mode option to setup the environment to install Bash.
5. After all the components are installed, you will need to restart your computer.
6. Next, open the **Control Panel**.
7. Click on **Programs**.
8. Select **Turn Windows features on or off**.

Continuing With Installation

9. Scroll down in the popup window and check the box for **Windows Subsystem for Linux**.
10. More components will install on your computer, after which select the popup **Restart Now**.
11. After reboot, open the **Windows Store** and search for **Ubuntu**. Click on Ubuntu and press install. After the program is installed, press **Launch**.
12. The command prompt will then open. You will see text that states the installation has begun. This can take a few minutes, so be patient. **Do not close the installation window!**
13. Afterwards you will be prompted to create a UNIX account. **Remember this info**. You will need the password for installing future programs and tools.

What to Look out For

During step 12, if you encounter the error code **0x8000000d**, open the Powershell in Admin mode by pressing the **Windows key + x** and selected Powershell(admin).

- ▶ Enter the following command:

```
Enable-WindowsOptionalFeature -Online -  
FeatureName Microsoft-Windows-  
Subsystem-Linux
```

- ▶ Go back to **Programs** in the Control Panel and select uninstall programs. Find Ubuntu and uninstall it. You can then continue from step 11.

Using the Bash Shell

- ▶ To access the bash shell tool, simply open a command prompt(or Powershell) and type **bash**.
- ▶ While it isn't designed to run Linux graphical applications, most UNIX commands will work, letting you run programs on the command line without the use of a virtual machine.

One More Thing

In order to install packages using the bash shell, we'll need to grab some updates. Use the following command:

```
sudo apt-get update
```

Installing Python

- ▶ Check to see if python3 is already installed using:

```
python3 --version
```

- ▶ If it is, great! Else, use the command:

```
sudo apt-get install python3
```

If for any reason you get stuck and can't continue you can follow along using this website: [IPython Online](#)

Pip

- ▶ **Pip** is a **package management system** that is used to install software packages that are written in Python. Pip's command line interface can make it extremely easy to install software through a simple command.
- ▶ To check if pip is already installed on your machine enter the command:

```
pip3 --version
```

If it is not installed, use the following:

```
sudo apt install python3-pip
```

- ▶ You will be prompted with a message that the installation will use disk space. Type '**Y**' and press enter.

Jupyter

- ▶ **Jupyter** is a open-source web application that makes it easy to write and share code, visualizations, and interactive output.
- ▶ It provides a python package that will allow you to host a local server that makes it simple to maintain your files and start working on some projects.
- ▶ To install, we will install using pip by using the following command:

```
sudo pip3 install jupyter
```

To run your server type the following on the command line:

```
ipython3 notebook
```

Your First Programs

- ▶ Let's try writing some actual programs in Python. After running your server and paste the link provided into your browser.
 - ▶ You can copy the link by **double clicking the text** and performing a **right click**. This will allow you to paste the link into your browser
- ▶ On the top right of your homepage click **New** and select **Python3**. A new notebook page where we can write programs should be opened in a new tab.

The Basics

Using the cells in the notebook page we can write executable code. For example if we type $2 + 2$ and click **Run**, we should see 4 print out.

If we use a **variable**, we can store that result for future use. Let's do the same thing but store the result in a variable called **sum**:

```
result = 2 + 2  
print(result)
```

Notice how we passed our variable into *print()*. Print is a function, which is a stored routine of code that we can call to perform a task.

Practice with Functions

Functions take **parameters** inside of the parentheses which it can use. For example, previously we gave *print()* our **sum** variable which it used to print our result to the screen.

By using the keyword **def** we can give our function a name and parameters. Let's create a function that can add two numbers:

```
def sum(num1, num2):  
    result = num1 + num2  
    print(result)
```

```
sum(2, 2)  
sum(10000, 25000)
```

Plots

Let's try using a function from a **library** to make a plot! By using the library **matplotlib** we can print a cool histogram of whatever values we choose. In this example we'll create a variable that can hold multiple numbers at once called an **array**. First use pip in the bash shell to install a new package for plotting:

```
sudo pip3 install matplotlib
```

Then in your jupyter notebook:

```
%matplotlib inline
import matplotlib
import numpy as np
mu, sigma = 100, 15
x = mu + sigma * np.random.randn(500)
plot = matplotlib.pyplot.hist(x, bins = 50)
```

Conclusion

- ▶ By now you should have an environment in which you can write Python code and save your work on your computer.
- ▶ We took a quick look at how to write a simple program and create a function that we can pass new values into.
- ▶ Jupyter will allow you to create more complex programs and create professional looking visualizations.
- ▶ By making use of these tools, you will find that there are many cases in your own studies that can make use of them.
- ▶ Contact me at: gesparza3@mail.csuchico.edu