Python 3.6 Environment

Python 3.6 Installation Linux (Ubuntu):

- Install the following dependencies:

```
$ sudo apt-get update

$ sudo apt-get install build-essential checkinstall

$ sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev
```

- Go to Download Python page on https://www.python.org/downloads/ (https://www.python.org/downloads/) and click Download Python 3.6.4 (You may see different version name).
- In the terminal, go to the directory where the file is downloaded and run the command:

```
$ tar -xvf Python-3.6.4.tgz
```

- This will extract your zipped file. Note: The filename will be different if you've downloaded a different version. Use the appropriate filename.
- Go to the extracted directory.

```
$ cd Python-3.6.4
```

- Issue the following commands to compile Python source code on your Operating system.

```
$ ./configure

$ make

$ make install
```

- Open Sublime text. To create a new file, go to File > New File (Shortcut: Ctrl+N).
- Save the file with .py file extension like: hello.py or first-program.py
- Write the code and save it (Ctrl+S or File > Save) . For starters, you can copy the code below:

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

- This simple program outputs "Hello, World!"
- Go to Tool > Build (Shortcut: Ctrl+B). You will see the output at the bottom of Sublime Text. Congratulations, you've successfully run your first Python program.

Install and Run Python in Windows

- Go to Download Python page on https://www.python.org/downloads/ (https://www.python.org/downloads/) and click Download Python 3.6.4 (You may see different version name).
- If your computer is running a 64-bit version of Windows, download the Windows x86-64 executable installer. Otherwise, download the Windows x86 executable installer. After downloading the installer, you should run it (doubleclick on it) and follow the instructions there.
- One thing to watch out for: During the installation you will notice a window marked "Setup". Make sure you tick the "Add Python 3.6 to PATH" checkbox and click on "Install Now".
- When Python is installed, a program called IDLE is also installed along with it. It provides graphical user interface to work with Python.
- Open IDLE, copy the following code below and press enter.

`print("Hello, World!")`

- To create a file in IDLE, go to File > New Window (Shortcut: Ctrl+N).
- Write Python code (you can copy the code below for now) and save (Shortcut: Ctrl+S) with .py file extension like: hello.py or your-first-program.py `print("Hello, World!")`
- Go to Run > Run module (Shortcut: F5) and you can see the output. Congratulations, you've successfully run your first Python program.

Install and Run Python using PyCharm

Refer https://www.jetbrains.com/pycharm/ (https://www.jetbrains.com/pycharm/)

Install and Run Python using Anaconda

Refer https://www.anaconda.com/download/ (https://www.anaconda.com/download/)

Introducton to Prompt Window

You now should see a white or black window that is waiting for your commands.

Prompt: Linux OS (Ubuntu)

If you're on Linux, you probably see `$` , just like this: `terminal`

```
$
```

To run Python script `test.py`

```
$ python3 test.py
```

For pip3:

```
$ sudo apt-get -y install python3-pip
```

```
$ pip3 install package_name
```

Prompt: Windows OS

On Windows, it's a `>` sign, like this: `command-line`

```
>
```

To run Python script `test.py`

```
> python test.py
```

For pip:

```
> pip install package_name
```

To check the installed versison of Python

For Windows

```
> python --version

Python 3.6.4
```

For Linux

```
$ python3 --version

Python 3.6.4
```

Using the Python Shell

Before starting to write programs, you ’ ll need to learn how to experiment with the Python shell. For now, you can think of the Python shell as a way to peer within running Python code. It places you inside of a running instance of Python, into which you can feed programming code; at the same time, Python will do what you have asked it to do and will show you a little bit about how it responds to its environment. Because running programs often have a context — things that you as the programmer have tailored to your needs — it is an advantage to have the shell because it lets you experiment with the context you have created.

In Windows cmd prompt

```
> python

>>>
```

To exit

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

In Linux Terminal

```
$ python3

>>>
```

To exit

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