



Global Knowledge®

Introduction to Python

Estimated time for completion: 45 minutes

Overview:

In this lab, you will install and configure Python and an IDE (or two).

Goals:

- Install Python 3
- Install Python 2 (if not already installed)
- Configure Python 3 to execute scripts
- Compare Python 2 and Python 3 behaviors

Notes:

- This lab exercise is meant for both OS X and Windows users. Certain steps include platform specific steps in pairs (one for Windows and one for OS X). Depending on your current OS, skip the other one in the instructions.

Part 1 – Install Python 3 and Python 2

At the time of writing, Python 3 is not installed by default in either OS X or Windows. You will download and install Python 3 now. You should prefer the 32-bit version.

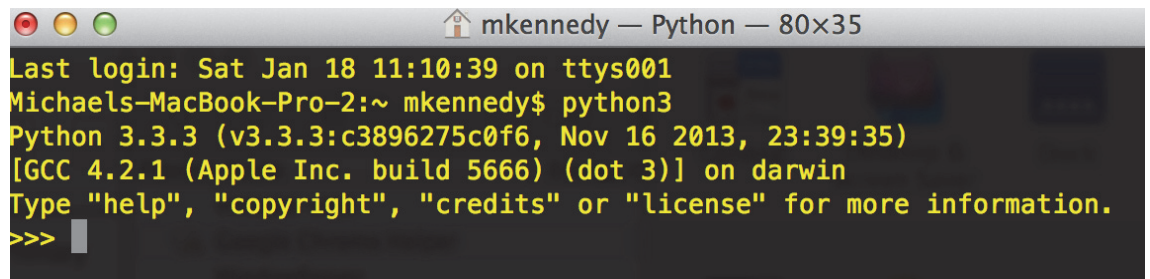
Please check with your instructor as they may have provided these already in cases of limited network availability.

1. Go to <http://python.org/download/> and download the latest version of Python 3. The easiest Windows download is **Windows x86 MSI Installer**. The easiest OS X version is **Mac OS X 32-bit i386/PPC Installer**.
2. **Windows users:** You will also need to download Python 2 from <http://python.org/download/>. Download **Python 2.7.* Windows Installer** as well.
3. OS X users already have Python 2 installed.
4. Execute the various installers.

- OS X should place Python 3 in the default location. An alias will be created at **/usr/local/bin/python3**.
 - Windows users may need to adjust their install location. You can accept the default location of C:\Python34 (varies by version). I prefer **C:\python\python34_32bit** so allow for multiple versions and both 32 and 64-bit variations.
 - Windows users install Python 2 at **C:\python\python27_32bit**.
5. Adjust the path
- **OS X users** should already have Python 3 in the path. Simply type **which python3** in a terminal windows to verify it was installed correctly. You should see that while the alias for python3 is located at **/usr/local/bin/python3** the actual install is probably somewhere like **/Library/Frameworks/Python.framework/Versions/3.4/bin/python3**.
 - **Windows users** may need to adjust the system path. You may also need to restart any command windows even if the path is correct. To set the path, right-click on your computer in Explorer, choose properties > advanced system settings > environment variables > Path. Add your Python install paths in the front of the path as follows (adjust for install variations):
C:\python\python34_32bit; C:\python\python27_32bit;C:\Other...
 Start a new command prompt and verify you have Python 3 located correctly. Type: **where python**. You should see the first directory above.

Part 2 – Distinguish Python 2 from Python 3

1. A *very common* problem newcomers to Python encounter is running the wrong version of Python. Because they are incompatible but similar, weirdness ensues. This step will help you distinguish and identify this problem.
2. There are two basic ways (we'll cover more later) to execute Python.
 - a. Typing python on the command line / terminal will launch the REPL. This typically has the version at the top, so just look there



```

mkennedy — Python — 80x35
Last login: Sat Jan 18 11:10:39 on ttys001
Michael's-MacBook-Pro-2:~ mkennedy$ python3
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 16 2013, 23:39:35)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>

```

- b. Execute a script directly, e.g. simply typing some_script.py in the command prompt / terminal. This is trickier because the file associations in Windows make this non-obvious

which version will execute the script.

We will examine this scenario both explicitly and by observing side-effects of the version differences.

Here are two pieces of information you will need to solve a version puzzle.

- i. **print** is a statement in Python 2 and is a method in Python 3. That means in Python 2 you write

```
print "Hello", "world"
```

and in Python 3 you write

```
print("Hello", "world")
```

- ii. Python has a data type called a tuple which is defined as

```
t = (1, 2, 3, 5, 8, 13, 21)
```

- c. Now create a file named `hello.py` and add two lines (the `#!` line can be omitted on Windows):

```
#!/usr/local/bin/python  
print("Hello", "Python")
```

- d. Run this script by simply typing its name and inspect the output. Note that if the script did not run, skip ahead to the associating file extensions and then return here.

Which version of Python did it run on? (Note: OS X users will need to make the file executable via **`chmod a+x ./hello.py`**). You can also explicitly invoke a python version (e.g. `python3 hello.py`). Play with this until you understand what is happening.

3. Next we will associate `.py` files in Windows and OS X.

- a. **Windows:** run the following two commands in an admin command prompt, be careful to adjust for version / path differences:

- i. **`assoc .py=Python.File`**

- ii. **`ftype Python.File=C:\python\python34\python.exe "%1" %*`**



```
c:\>assoc .py=Python.File
.py=Python.File

c:\>ftype Python.File=C:\python33\python3.exe "%1" %*
Python.File=C:\python33\python3.exe "%1" %*

c:\>hello.py
Looks like you have file associations working with Python scripts!

c:\>_
```

- b. **OS X / Linux:** Unix-based systems use a 'shebang' line at the top of the file. You need to simply add the correct one. This is frequently:

`#!/usr/local/bin/python3`

Scripts also need to be executable, so use the shell command **`chmod a+x filename.py`**

Afterwards, you should be able to execute the script (may need local file reference `./filename.py` vs. `filename.py`).

- 4. Now you have a chance to explore the REPL. Simply type `python` (or `python3`), you'll want to make sure you get the Python 3 REPL running.
 - a. Type `help()` to see the help options. Explore some of the suggestions like modules, etc.
 - b. Try creating a variable and working with it across lines in the REPL. You can also define multi-line objects such as:

```
lookup = {
    "name": "Joe",
    "age": 27
}
```

- c. Spend some time and play around here.
- 5. Finally, it's time to install some IDEs.
 - a. You should install **PyCharm**, start by downloading it here:

<http://www.jetbrains.com/pycharm/>

Get the professional trial if possible.

- b. If you have Visual Studio installed, you may also want to try **Python Tools for Visual Studio**, you can install it from here:

<https://pytools.codeplex.com/>

- c. You may also want something more lightweight, Sublime Text is good for this too.

<http://www.sublimetext.com/>