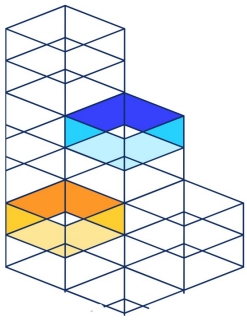


Data Science Introduction & Setup Checklist

As prepared by GS Jackson

Package Manager Installations Regardless of Operating System

Installing and Usage of Conda, Anaconda & Jupyter Notebooks

ANACONDA/CONDA is a editor and emulator framework that includes Jupyter Notebooks that acts as a wiki / complier for common programming languages and tools. It also includes a variety of other modules using around Big Data & Data Science.

For all operating systems you can download Anaconda or the mini version Conda at - https://www.continuum.io/downloads

The **Homebrew** (Mac OS, Linux) and **Chocolatey** (Windows) installation is below.

Installing Data Science Applications & Tools via Scripting Frameworks (Brew & Chocolatey)

Mac OS / Linux

The following programs are for installation and configuration on Mac OS X and Linus around work with Data Science. They include the frameworks for Big Data / Data Science including the programming languages necessary to configure.

For Mac OS X we will be using a scripting program called Homebrew.

The equivalent for Linux is called Linuxbrew.

Homebrew

Install Homebrew on MAC OS X, LINUX go to the website - http://brew.sh/

To install Homebrew –

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install.sh)
"
```

If you get an error in the installation –

```
sudo chown -R $USER /usr/local; brew update
```

```
brew install wget
```

All installation locations are here -

```
$ cd /usr/local
```

```
$ find Cellar

Cellar/wget/1.16.1

Cellar/wget/1.16.1/bin/wget

Cellar/wget/1.16.1/share/man/man1/wget.1


$ ls -l bin

bin/wget -> ../Cellar/wget/1.16.1/bin/wget
```

Brew uses modules of programs called formulas. You can see view a list of these formulas here - http://brewformulas.org/

Install Git Using Brew

```
brew install git
```

Install Anaconda

```
brew cask install anaconda
```

Install Python, Numpy, Scipy, Matplotlib

```
brew tap homebrew/science && brew install python numpy
scipy matplotlib
```

You can learn more about these at their respective repositories:

homebrew/science

samueljohn/python

Install Ipython and Corresponding Packages via Brew & PIP

```
brew install zeromq
pip install jinja2
pip install tornado
pip install pyzmq
pip install ipython
```

Install Pandas

```
pip install pandas
```

Testing

Open up Ipython

```
import numpy
import scipy
import matplotlib
import pandas
```

Installing iJavascript for Jupyter Notebooks for Mac OS

iJavascript

```
ruby -e “$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)”
brew install pkg-config node zeromq
sudo easy_install pip
pip install --upgrade pyzmq jupyter
npm install -g ijavascript
ijsinstall
```

To discover what programming languages and modules have been configured with Jupyter you can issue the following command:

```
jupyter kernelspec list
```

De-install Homebrew -

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/uninstall.s
h)"
```

CHOCOLATEY

Windows Only

Install Windows Powershell

The PowerShell team has also put together a script you can invoke directly from PowerShell. It’s a one-liner cmdlet that downloads and runs the setup wizard automatically. All you have to do is paste the code snippet in and hit the Enter key.

Fire up PowerShell and copy/paste the following cmdlet into the window:

```
iex "& { $(irm https://aka.ms/install-powershell.ps1) } -UseMSI"
```

Install Chocolatey

Running PowerShell as an Administrator cut and paste this into the command line -

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex
((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.
ps1'))
```

If you don't see any errors, you are ready to use Chocolatey!

```
choco
```

Install Git Using Chocolatey

```
choco install git.install
```

Install Anaconda Using Chocolatey

```
choco install anaconda3
```

Install Python

```
choco install python
```

Install Numpy

```
choco instal numpy
```

Installation and Usage of Github Regardless of MacOS, Windows, or Linux

Signup for a GITHUB account –
https://github.com/

Our Tech Talent South Github location is -
https://github.com/enterlifeonline/techtalentsouth

Pull Github TTS Data Science Repository

Initialized it with a LICENSE and README.md

Create a local directory called GITHUB
Created a local directory called REPOSITORIES

Within the `./Repositories` directory

From the local command line:

```
git clone https://github.com/enterlifeonline/techtalentsouth.git

git config –global user.name "Your Name"
git config –global user.email your.email.domain.com
```

Then from the local command line:

```
git status
```

–
When you want to update your Tech Talent South repository -

Within the `./Repositories` directory

```
rm -rf ./TechTalentSouth

git clone https://github.com/enterlifeonline/techtalentsouth.git
```

Set Up Sublime Text 3 to use Python 3

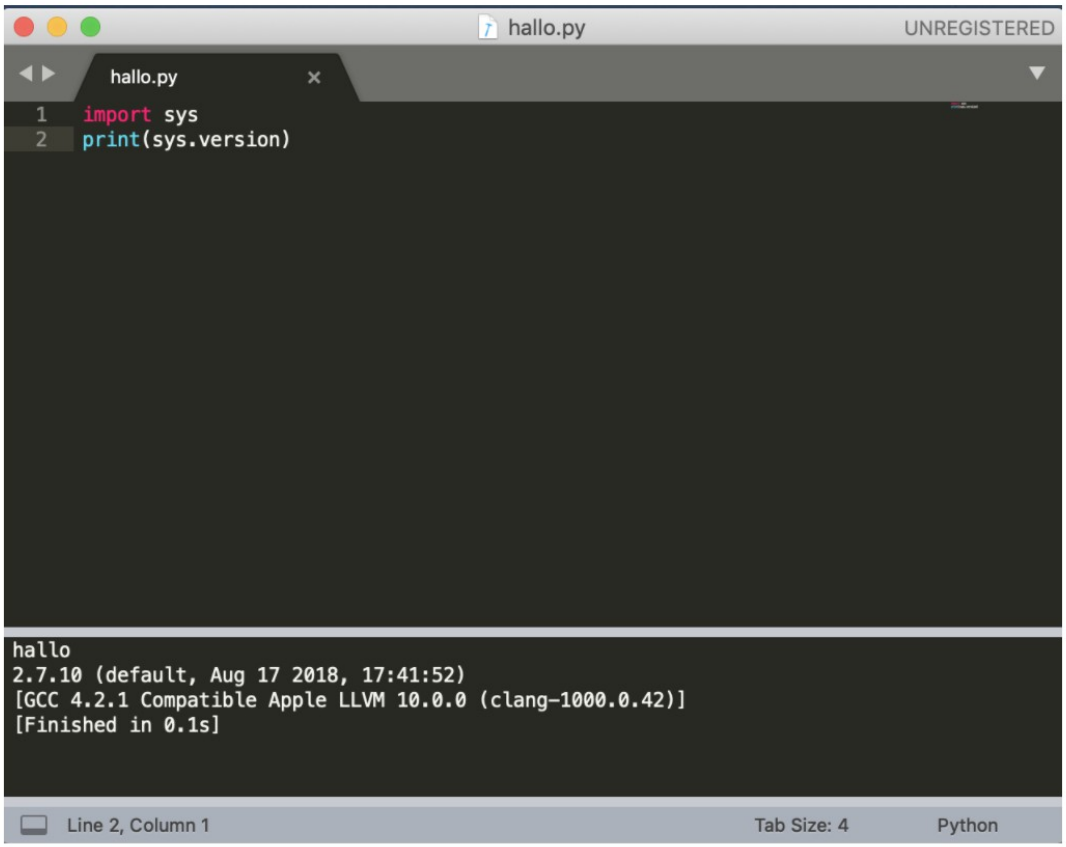
Download and Install Sublime 3 - https://www.sublimetext.com/3

Create a `new file` and save it with extension `.py` for example save it as `checkversion.py`

Go to `Tools -> Build System -> Python` then type on your `checkversion.py`

```
import sys
print(sys.version)
```

then press `CTRL + B` to run the code on Sublime



Example check version on Sublime Text 3

> ***First:*** *Check where is your Python3 path using this command*

UNIX STYLE

```
$ which python3
/usr/local/bin/python3
```

WINDOWS STYLE

```
C:\Users\<users>py
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915
32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more
information.
>>> import os
>>> import sys
>>> os.path.dirname(sys.executable)
'C:\\Users\\<user>\\AppData\\Local\\Programs\\Python\\Python37-32'
```

> ***Second:*** *Add new build system on your Sublime*

`Tools -> Build System -> New Build System` and make sure that the new build system has this following command

UNIX STYLE

```
{
 "cmd": ["/usr/local/bin/python3", "-u", "$file"],
 "file_regex": "^[ ]File \"(...?)\", line ([0-9]*)",
 "selector": "source.python"
}
```

WINDOWS STYLE

```
{
 "cmd":["C:/Users/<user>/AppData/Local/Programs/Python/Python37-
32/python.exe", "-u", "$file"],
 "file_regex": "^[ ]File \"(...?)\", line ([0-9]*)",
 "selector": "source.python"
}
```

and save this file as `newPython3.sublime-build`

> ***Third:*** *Select your new system build* `newPython3` *and re-run the* `checkversion.py` *and now it should be using* `Python 3`