Python Data Science Course - Week 1 - Day 1

Objectives for this week:

- Installing Python, Basic Terminal Commands
- Data Types, Variable Declation
- Strings / String methods
- Ints vs. Floats / Math in Python

Todays Agenda

- Course introduction
- Learning Basic Terminal Commands
- Installing Python
- Learning to use Jupyter Notebooks

Course Structure.

Phase 1 - Programming Fundamentals and Working with Web Data

Phase 2 - Data Analysis and Statistical Analysis

Phase 3 - Python Web Dashboard Development

Class Structure

Each day will consist of:

- 10 minutes: Meditation and Centering
- 30 minutes: Daily Code Challenge Presentations
- 70 minutes: Lecture/Code Along
- 90 minutes: Lab/Review
- 10 minutes: Wrap up/Review

Homeworks and Projects

Homework will be assigned every Wednesday. Due Monday, Tuesday, Wednesday of the following week.

6 projects (3 week long and 3 multiweek)

12 homeworks

Using Jupyter Notebooks

Usage

To run the selected code cell, hit

`Shift + Enter`

In [ ]:

```
# jupyter magic functions have % prepended

%magic
```

In [ ]:

```
%magic??
```

HTML and CSS

example of html

Javascript

In [ ]:

```
%%javascript
alert("hello world")
```

Python statements

In [ ]:

```
# EXAMPLES
print("hello world")
```

Terminal commands

In [ ]:

```
!ls -la
```

Command Line Basics

Useful Commands:

`pwd` `cd` `ls` `cat` `echo`

```
$ pwd
    /directory/you/are/in
```

```
In [ ]: !pwd
```

```
In [ ]: !cd
```

```
In [ ]: !cd ../week-001/
```

```
In [ ]: !ls
```

```
In [ ]: !ls -la
```

```
In [ ]: !cat README.md
```

```
In [ ]: !echo "hello"
```

Python Basics

Python Language Introduction

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.[28]

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.[29]

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming[49] and metaobjects (magic methods)).[50] Many other paradigms are supported via extensions, including design by contract[51][52] and logic programming.

source (https:/https://en.wikipedia.org/wiki/Python_%28programming_language%29)

What is Python good for?

- High level programming and doing a lot with minimal code
- Readable language created with linguistics in mind
- Popularity and widespread support
- Integration with other languages/software
- Object Oriented paradigm

Resources For Python

https://docs.python.org/3/ (https://docs.python.org/3/)
https://www.fullstackpython.com/ (https://www.fullstackpython.com/)
https://awesome-python.com/ (https://awesome-python.com/)
https://automatetheboringstuff.com/ (https://automatetheboringstuff.com/)
https://www.twilio.com/quest/learn/python (https://www.twilio.com/quest/learn/python)
https://www.cs.hmc.edu/twiki/bin/view/CSforAll (https://www.cs.hmc.edu/twiki/bin/view/CSforAll)
https://www.grahamwheeler.com/posts/python-crash-course.html (https://www.grahamwheeler.com/posts/python-crash-course.html) http://greenteapress.com/thinkpython/html/index.html (http://greenteapress.com/thinkpython/html/index.html) https://pymbook.readthedocs.io/en/latest/ (https://pymbook.readthedocs.io/en/latest/) https://techdevguide.withgoogle.com/ (https://techdevguide.withgoogle.com/) https://python.swaroopch.com/ (https://python.swaroopch.com/) https://leetcode.com/ (https://leetcode.com/)
https://projecteuler.info (https://projecteuler.info)

Invoking the python interpreter

Scripts/Programs

The simplest script is just a file with top level (0 indent) code.

example.py:

```
line = "hello world"
     print(line)
```

result:

```
$ python example.py
    "hello world"
```

The most simple commonly used script format is:

```
#/usr/bin/env python

    def main():
        print("hello world")

    if __name__ == "__main__":
        main()
```

result:

```
$ python example2.py
    "hello world"
```

Interactive using IDLE, ipython or jupyter notebook.

Similar to using the command line, interpreters handles python code statement by statment. Allowing you to run python code as such:

```
>>> text = "hello world"
     >>> print(text)
```

Lab

Installing Python Environment

On Windows:

1. Download Anaconda from official page here (https://www.anaconda.com/download/#windows)
2. Install Anaconda using these steps here (https://docs.anaconda.com/anaconda/install/windows/)
3. Open "Jupyer Notebook" from your command prompt.

On Mac:

Using Homebrew:

1. `brew install pyenv`
2. `pyenv install -l`
3. `pyenv install 3.8.5`
4. `echo 'eval "$(pyenv init -)"' >> ~/.bash_profile`
5. `pyenv global 3.8.5`
6. `brew install jupyter`
7. `jupyter notebook`

Using Anaconda

1. Download Anaconda from official page here (https://www.anaconda.com/download/#macos)
2. Install Anaconda using these steps here (https://docs.anaconda.com/anaconda/install/mac-os/)
3. Open "Jupyer Notebook" from your terminal.

On Linux:

Using Ubuntu 20.04 or later:

1. `sudo apt install python3-notebook jupyter jupyter-core`

Using Ubuntu 18.04-19.10:

1. `sudo apt install python3-notebook jupyter jupyter-core python-ipykernel`

Using Anaconda:

1. Download Anaconda from official page here (https://www.anaconda.com/download/#windows)
2. Install Anaconda using these steps here (https://docs.anaconda.com/anaconda/install/windows/)
3. Open "Jupyer Notebook" from your command prompt.

PIP

1. Open `terminal` , `command prompt` or other command line application.
2. `pip install jupyter`
3. `jupyer notebook`

In [ ]:

```

```