Introduction to Python and Jupyter Notebook

February 23, 2018

1 Table of Contents

- Section 2
  - Section 2.1
  - Section 2.2
  - Section 2.3
    - Section 2.3
    - Section 2.3
- Section 3
  - Section 3.1
  - Section 3.2
- Section 4
  - Section 4.1
  - Section 4.2
    - Section 4.2.1
    - Section 4.2.2
    - Section 4.2.3
- Section 5
- Section 6
- Section 7

2 Introduction

2.1 What is Python and Jupyter Notebook?

Python.org:

Python is a programming language that lets you work quickly and integrate systems more effectively.

Jupyter.org:

> Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

2.2 Why should we know them?

In academic and research we can use Python and its lots of labraries to do wide range of things from basic calculation to complex statistical methods including machine learning. Since it is open-source, it is free to use.

2.3 Where can I get it?

There are many ways to install Python and its libraries. However, we are going to guide you in a pretty stable way, Anaconda.

Anaconda Cloud Anaconda is a platform for Python that bundles most of the essential packages required in research. You can download it from here. **Note that we will use Python version 3.x in this course.**

Anaconda.org:

> Anaconda Cloud is where data scientists share their work. You can search and download popular Python and R packages and notebooks to jumpstart your data science work.

Package Managers We do not need to install any additional packages in this class. However, when you need it in the future recall these two names: `pip` and `conda`. For more information you can check this out.

3 Jupyter Notebook

3.1 How to start Jupyter Notebook server?

To use Jupyter Notebook (which you are using) you would need to have basic knowledge in command line interface (cli). Here is an example on how to start Jupyter Notebook from start.

- Open Terminal (Mac/Linux) or Command Prompt (Windows).
- `cd` into project folder where your `.ipynb` file live.
- Run `Jupyter Notebook` in that folder.

You should see something like this:

```
C:\Users\username> mkdir labclass
C:\Users\username> cd labclass
C:\Users\username\labclass> jupyter notebook
[I 10:26:00.076 NotebookApp] Serving notebooks from local directory: /Users/username/labclass
[I 10:26:00.076 NotebookApp] 0 active kernels
[I 10:26:00.076 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=
```

```
[I 10:26:00.076 NotebookApp] Use Control-C to stop this server and shut down all kernels (twic
[C 10:26:00.076 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

3.2 How to use it?

This section will list the most used command shortcuts. For full version of docs you can visit The Jupyter Notebook documentation.

- `arrows` : *move cursor*
- `enter` : *enter edit mode*
- `esc` : *quit edit mode*
- `y` : *change cell to code mode*
- `m` : *change cell to markdown mode*
- `ctrl` + `enter` : *run selected cell*
- `shift` + `enter` : *run selected cell and move cursor to the next cell*
- `h` : *open help panel, list of shortcuts*

4 Python

4.1 Built-in functions and basic control flow

Here we will try to write some Python built-in functions to get how it works. This is brought from official Python website. For more information follow the official tutorial.

```
In [ ]: # print function
        print("Hello, I'm Python!")
```

```
In [ ]: # arithmatic
        print(1+3) # plus
        print(2-1) # minus
        print(3*6) # multiply
        print(2**3) # power
        print(17/3) # divide
        print(17//3) # floor divide
```

```
In [ ]: # compound data
        # list
        fruits = ['Banana', 'Apple', 'Lime']
        loud_fruits = [fruit.upper() for fruit in fruits]
        print(loud_fruits)

        # list and the enumerate function
        print(list(enumerate(fruits)))
```

In [ ]:
```
# conditions
x = -1

if x < 0:
    x = 0
    print('Negative changed to zero')
elif x == 0:
    print('Zero')
elif x == 1:
    print('Single')
else:
    print('More')
```

In [ ]:
```
# for loop
words = ['cat', 'window', 'defenestrate']
for w in words:
    print(w, len(w))

# Loop over a slice copy of the entire list.
# if length of being looped word longer than 6
# insert this word to index 0 of the list
for w in words[:]:
    if len(w) > 6:
        words.insert(0, w)
print("inserted list:", words)
```

In [ ]:
```
# for loop 2
for i in range(5):
    print(i)

a = ['Mary', 'had', 'a', 'little', 'lamb']
for i in range(len(a)):
    print(i, a[i])
```

In [ ]:
```
# while loop
x = 0
while x < 10:
    print(x)
    x += 1
```

In [ ]:
```
# functions (methods)
# Python 3: Fibonacci series up to n
def fib(n):
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()
```

```
fib(1000)
```

4.2 Mathematics and Statistics packages

In research we will inevitably have to do some statistics. It is good to know how we can do mathematics and statistics operation with Python. As we are using Anaconda which already installed most required packages, so you do not need to install additional package here.

- The packages we will learn today are `numpy`, `pandas` and `matplotlib`.

4.2.1 NumPy

NumPy is a python library to help dealing with mathematics especially array. As it can do so many things, the examples listed below may not be able to cover all. For more information, you can visit NumPy's official tutorial.

In [ ]:
```
import numpy as np

# create array of range [0, 15)
a = np.arange(15)
print("basic array:", a)

# reshape array
print("\nreshaped:", a.reshape([3, 5]))

# or do it at once
print("\nall at once:", np.arange(15).reshape([3, 5]))
```

In [ ]:
```
# create array from list
mylist = [1,2,3,4]
print("array from list:", np.array(mylist))

# create 2d array
print("\n2d array:", np.array([mylist, mylist]))
print("\ncheck dimension:", np.array([mylist, mylist]).shape, "<- 2 rows 4 columns")
```

In [ ]:
```
# create array of ones
print("array of ones:", np.ones([3,3]))

# create array of zeros
print("\narray of zeros:", np.zeros([5,5]))

# some constants
print("\npi:", np.pi)
print("e:", np.e)

# basic arithmetics
print("\nsum, 1+2+3+4+5:", np.sum([1,2,3,4,5]))
print("power, 2^8:", np.power(2, 8))
```

4.2.2 pandas

pandas is an easy-to-use python library for data structures and data analysis. pandas implements a number of statistical functions that can be used in research. You can learn more about pandas in these pandas's official quick start tutorials:

- 10 mins tutorials
- pandas tutorials

```
In [ ]: import pandas as pd

        df_for_line = pd.DataFrame(np.random.rand(100, 1), columns=['col1'])
        df_for_scatter = pd.DataFrame(np.random.rand(5,2), columns=['col1','col2'])
        df_for_bar = pd.DataFrame(np.random.rand(5, 1), columns=['col1'])

        print(df_for_scatter)

        # create new dataframe by concatenate existing ones
        new_df_concat_down = pd.concat([df_for_scatter[3:], df_for_scatter[3:]]) # use [3:] to
        new_df_concat_side = pd.concat([df_for_scatter, df_for_scatter], axis=1)

In [ ]: # we can also see dataframe rendered as HTML
        df_for_scatter

In [ ]: # notice index start from 3 instead of 0
        new_df_concat_down

In [ ]: new_df_concat_side
```

4.2.3 Matplotlib

Matplotlib is a popular Python 2D plotting library. You can do most of data visualization using Matplotlib. pandas plotting methods also use Matplotlib as a based tool.

- Pyplot tutorial by matplotlib.org
- Matplotlib tutorial by Nicolas P. Rougier

```
In [ ]: import matplotlib.pyplot as plt
        %matplotlib inline

        plt.plot(df_for_line)
        plt.title('Line plot')
        plt.figure()

        plt.scatter(df_for_scatter.col1, df_for_scatter.col2)
        plt.title('Scatter plot')
        plt.figure()

        plt.bar(df_for_bar.index, df_for_bar.col1)
        plt.title('Bar plot')
```

```
plt.figure()

from skimage import data

plt.imshow(data.chelsea())
```

5 Assignment

- Click here to open assignment notebook.

6 FAQs

- **How should I search when I am stuck?**
  - You can use google with specific keywords. For example, if you are doing some table processing using `pandas`, you should include `pandas` in keyword instead of `python`.
- **What sources from search results should I set higher priority to check out?**
  - This can be various, but first top focus should include **StackOverflow**, **GitHub**, and official sites of Python or packages in question.

7 Learning Materials

- Official Python tutorial
- Chula's Python 101