BU.330.740 Large Scale Computing with Hadoop

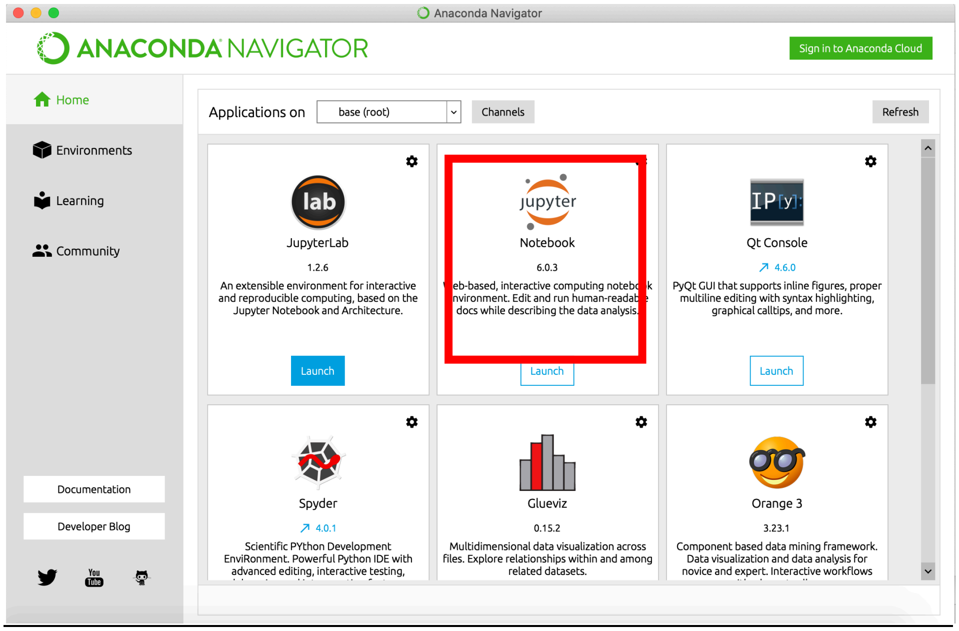
Lab 1. Python Tutorial

Learning Goal: practice python basics

We will use Anaconda, a popular Python/R data science platform for Python programming in this class.

1. Download and install Anaconda from https://www.anaconda.com/products/individual according to your own operating system. For Windows users, if you encounter problems during installation, please refer to https://docs.anaconda.com/anaconda/install/windows/ for some common troubleshooting issues.

2. Open Anaconda, and we will use Jupyter Notebook for the first lab.



JupyterLab: An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook: eb-based, interactive computing notebo k nvironment. Edit and run human-readab e docs while describing the data analysis.

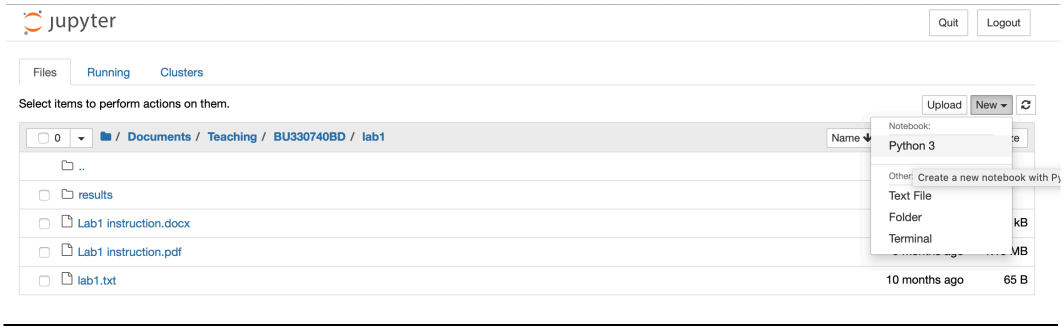
Qt Console: PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder: Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing,

Glueviz: Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3: Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows

3. Once the webpage is launched, you can navigate to your working folder, and make a new Python 3 notebook.



Select items to perform actions on them.

0 / Documents / Teaching / BU330740BD / lab1

..

results

Lab1 instruction.docx

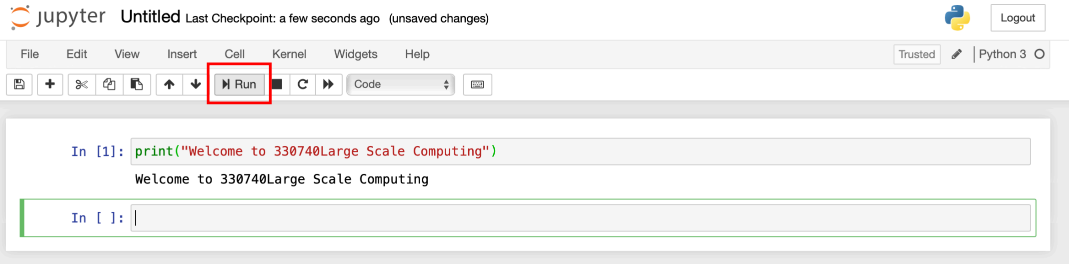
Lab1 instruction.pdf

lab1.txt 10 months ago 65 B

4. As the first step of learning any other programming language, let’s first try the print function of Python. In a cell, type in

   ```
   print("Welcome to 330740Large Scale Computing")
   ```

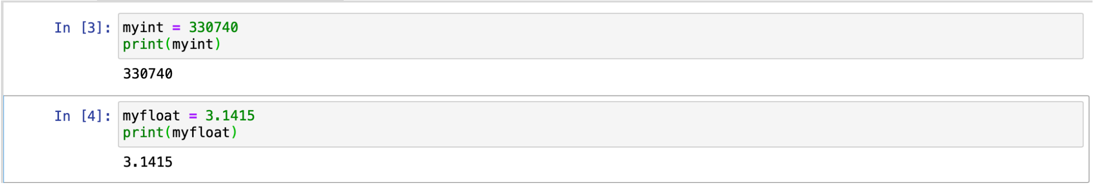
   and then click “Run”.



```
In [1]: print("Welcome to 330740Large Scale Computing")

        Welcome to 330740Large Scale Computing

In [ ]: |
```

5. Python supports two types of numbers - integers and floating point numbers. In the next cell(s), type in the following and run

   ```
   myint = 330740
   print(myint)
   myfloat = 3.1415
   print(myfloat)
   ```



```
In [3]: myint = 330740
        print(myint)

        330740

In [4]: myfloat = 3.1415
        print(myfloat)

        3.1415
```

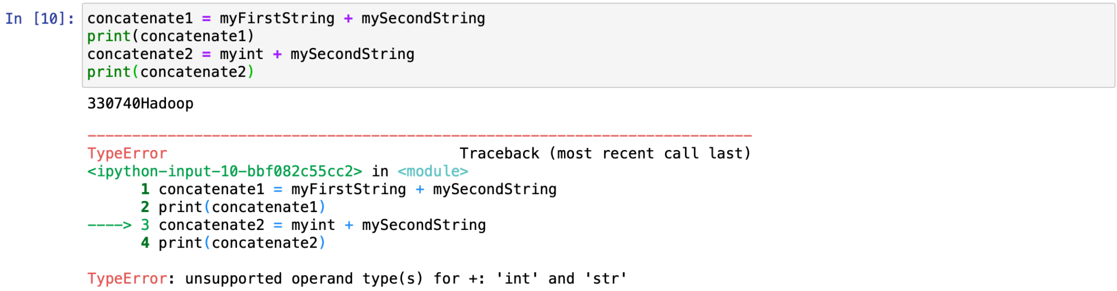
6. Strings are defined either with a single quote or a double quotes. In the next cell, type in the following and run

   ```
   myFirstString = '330740'
   print(mystring)
   mySecondString = "Hadoop"
   print(mystring)
   ```

7. Then let’s do some operations on these numbers and strings. In the next cell(s), type in the following and run

   ```
   addition = myint + myfloat
   print(addition)
   subtraction = myint – myfloat
   print(subtraction)
   multiplication = myint * myfloat
   print(multiplication)
   division = myint / myfloat
   print(division)
   mod = myint % myfloat
   print(mod)

   concatenate1 = myFirstString + mySecondString
   print(concatenate1)
   concatenate2 = myint + mySecondString
   print(concatenate2)
   ```

   What do you find out?



```
In [10]: concatenate1 = myFirstString + mySecondString
         print(concatenate1)
         concatenate2 = myint + mySecondString
         print(concatenate2)

         330740Hadoop

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         <ipython-input-10-bbf082c55cc2> in <module>
               1 concatenate1 = myFirstString + mySecondString
               2 print(concatenate1)
         ----> 3 concatenate2 = myint + mySecondString
               4 print(concatenate2)

         TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

8. Python supports arrays as lists. Next let's define a list of numbers, and a list of strings. In the next cell(s), type in the following and run

```
intList = [330, 740]
stringList = []
stringList.append("large")
stringList.append("scale")
stringList.append("computing")
print(intList[0])
print(intList[1])
print(stringList)
```

9. Python uses boolean variables to evaluate conditions. The boolean values True and False are returned when an expression is compared or evaluated. In the next cell, type in the following and run

```
name = "John"
age = 23
print(name == "John")
print(age > 25)
```

Python has if conditional statement, and use elif for elseif. Type the following example in the next cell and run.

```
score = 85
if score >= 90:
      print("A")
elif score >=80:
      print("B")
else:
      print("C")
```

10. Let's try some useful for loops in Python. In the next cell(s), type in the following and run

```
for x in range(5):
    print(x)

for x in stringList:
    print(x)
```

11. Functions in python are defined using the block keyword "def", followed by the function's name and arguments if any. Define a function for me to covert total score to letter grade as following:

```
def convert2letter(totalScore):
    if totalScore >= 90:
        return "A"
    elif totalScore >= 80:
        return "B"
    else:
        return "C"
```

and then test it using

```
johnScore = 80
print(convert2letter(johnScore))
```

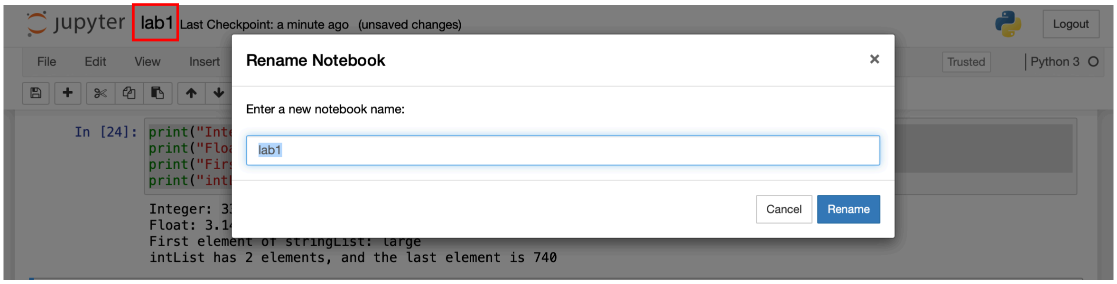
12. Python uses C-style string formatting to create new, formatted strings. The "%" operator is used to format a set of variables. See some examples below.

```
print("Integer: %d" % myint)
print("Float: %f" % myfloat)
print("First element of stringList: %s" % stringList[0])
print("intList has %d elements, and the last element is %d" % (len(intList),intList[-1]))
```

13. A dictionary is a data type similar to arrays, but works with keys and values instead of indexes. For example, I can maintain a students' scorebook in the following example.

```
scoreBook={}
scoreBook["John"]=80
scoreBook["Jack"]=75
scoreBook["Jill"]=92
print("Jack receives %s from my course" % convert2letter(scoreBook["Jack"]))
print("Jill receives %s from my course" % convert2letter(scoreBook["Jill"]))
```

14. Finally, let's rename/save the notebook. Click the name of the notebook (as shown below), the "Rename Notebook" interface will show.



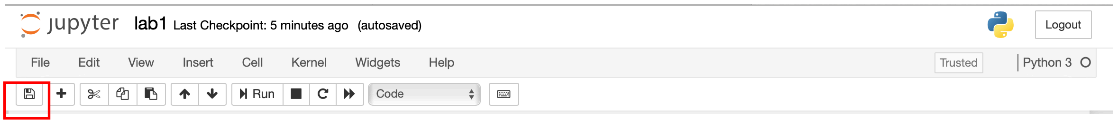
```
In [24]: print("Inte
         print("Floa
         print("Firs
         print("int

         Integer: 3
         Float: 3.1
         First element of stringList: large
         intList has 2 elements, and the last element is 740
```

Rename Notebook ×

Enter a new notebook name:

lab1

Also, use the disk icon to save your notebook.



15. For more examples, please explore https://www.learnpython.org/en/Welcome.