Welcome to BST 281 Lab 2

7 Feb, 2019

Mike MacArthur

macarthur@g.harvard.edu

Office Hours: Fridays 9-10a

FXB 2nd floor atrium

No office hours this Friday

Homework

First homework due Monday 2/11 by 11:59p on Canvas

Extensions are allowed but must be approved ***in advance***
(The day before is not in advance)

Homework can be downloaded and submitted from the Assignments section of Canvas

Lab Agenda

1. Cluster Tutorial
    i. Thanks Francesco!
2. Simple Python Commands
3. Simple Command Line
4. Open Questions

Python Practice

1. Open python

- `python`

2. Test the following commands
   - `print("Nice to meet you")`
   - `name = "Mike"`
   - `print("Nice to meet you %s" %name)`
3. Test some simple mathematical commands:
   - `a = 1`
   - `b = 2`
   - `c = a + b`
   - `print("What is the sum of a+b? %d" %c)`

Next we will run all of these commands at once by making them into a Python script.

1. In Atom copy and paste each of these lines into a new file
2. Save the file as "pythonTest.py"
   - Note the directory that you saved in it!
3. In the command line navigate to the directory containing your new file and run it
   - `python pythonTest.py`

** Command line practice

1. Log in to the cluster
   - You are now in your home directory
2. List the entries in your home directory (maybe it is empty!)
3. Make an empty .txt document and save it in your home directory
4. List the entries in your home directory again confirm that your new document is there.
5. Make a new directory called "BSTpractice" in your home directory
6. Move your empty .txt file into your new directory.
7. Navigate into the new directory and confirm that your empty .txt file is there

```
def mult_fun(a, b):
    return a * b

print(mult_fun(5, 2))
print(mult_fun('z', 10))
```

- Now add a doctest section to the function

```
def mult_fun(a, b):

    """
    Multiplies two inputs

    :param  a:    first input to multiply
    :type   a:    num or str
    :param  b:    second input to multiply
    :type   b:    num or str
    :returns:                   num or str -- product of two inputs

    >>> mult_fun(3, 2)
    6

    >>> mult_fun(3, 'b')
    'bbb'

    """
```

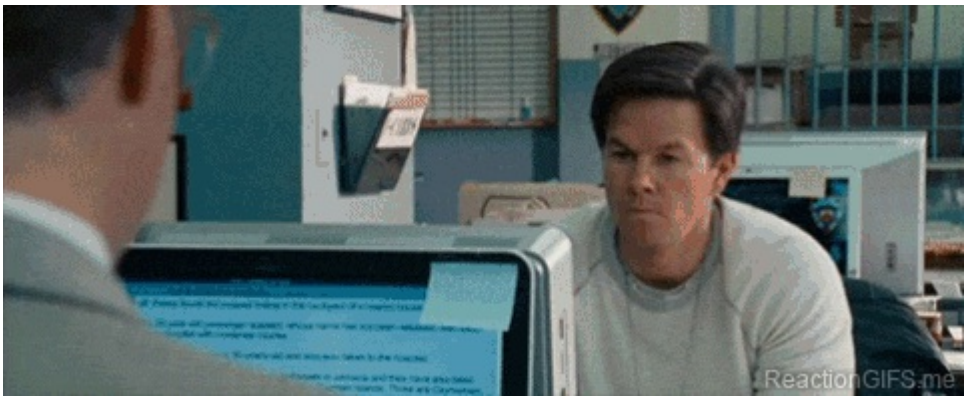
```
    return a * b

print(mult_fun(5, 2))
print(mult_fun('z', 10))
```

- Finally, let's doctest our new script
  `python -m doctest -v newFunction.py`

Pretty cool

Troubleshooting and Python practice!

Canvas Lab 2 Page



Download the lab02_practice.py file from Canvas

- Start a python instance in your terminal
  - `python`
- Run the commands in python
  - Fix any errors that are throw n w hen running the commands
- Using the python interpreter answ er the follow ing questions

    i. How many elements are in **aiX**
    ii. What is the value of the last element of **aiX**
    - Remove the last element of **aiX**
    i. What are the value of the first three elements of **aiX**
    - Change the value of the second element to 100
    i. Generate a variable **iXSum** which is the sume of all elements in **aiX**
    ii. Create a command that will print the value of **iXSum** if it is greater than 0, or display a message if it is not
    iii. Create a new variable **aList** which is the sum of all elements in **aiX** and **astrY**
    iv. Create a new variable **strMySting** where the 12 in **strNewString** is replaced with any other number
    v. Create a new variable **aiKeys** which consists of the keys of **hZ**
    vi. Create a new variable **aiSortedKeys** which consists of the sorted keys of **hZ**

Making commands into a Python script

1. Open a new script and call it lab01_script.py
2. (Optional) Optimize the script with a "shebang" line
3. Make a docstring as the next lines in your file

```
"""
<your name>
<today's date>
"""
```

4. Make a new block of code starting with `if __name__ = "__main__":`
5. In this block, create a variable `strMessage = "Hello, World!"` and print the message
6. Save the program and run it from the terminal
    - `python lab01_script.py`
7. In your script, under the `__name__ = "__main__"` block, create a variable called `strName` which stores your name as string
8. Make a function called `funcGreet` above the `__name__ = "__main__"` which takes a string input and prints "Hello, 'string'!"
9. Add `funcGreet(strName)` to the `__name__ = "__main__"` block
10. Run your script, you should see two outputs
11. Create another function called `funcDivSum` that takes two lists as an input and returns the ratio of their sums. Since division by zero is not allowed the function could raise an exception if any sums are zero.
12. Inside the `__name__ = "__main__"` block, make two lists, `aList1` and `aList2`. The sum of one of the should be zero. Create a variable `dRatio` which is the result of calling the function with those two lists and print it.
13. Save the script and run it
    - `python lab01_script.py`
14. Finally, to back to the script and make sure that neither `aList1` nor `aList2` have a sum of 0. Save the script and run it again: you should see three outputs.