Welcome to BST 281 Lab 2

1 Feb, 2018

Mike MacArthur

macarthur@g.harvard.edu

Office Hours: Fridays 2-3p

Kresge Student Lounge

No office hours this Friday

Homework

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

I will be away on 2/2 so send questions by Thursday 2/1!

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

Homew ork can be dow nloaded and submitted from the Assignments section of Canvas
Canvas Assignments page

Lab Agenda

1. Review Python doctest
   i. Test on homew ork
   ii. Create new function w ith doctest
2. Troubleshooting practice
   i. Perform practice exercise
   - lab02_practice.py file
3. Open time
   - w ork on Biological Sequences Jupyter Notebook or homew ork

---

---

doctest in Python

The doctest module searches for pieces of text that look like interactive Python sessions, and then executes those sessions to verify that they w ork exactly as show n. There are several common w ays to use doctest:

- To check that a module's docstrings are up-to-date by verifying that all interactive examples still w ork as documented.
- To perform regression testing by verifying that interactive examples from a test file or a test object w ork as expected.
- To w rite tutorial documentation for a package, liberally illustrated w ith input-output examples. Depending on w hether the examples or the expository text are emphasized, this has the flavor of "literate testing" or "executable documentation".

(From Python documentation)

We w ill use doctest to ensure that code edited in homew ork is functioning as intended

Executing a doctest:

Run the follow ing code:

`python -m doctest -v p01-introduction.py`
OR
`python -m doctest p01-introduction.py`
What is the difference w hen -v is removed

Without editing the homew ork file, this is the end of the doctest output:

```
2 items had no tests:
    p01-introduction
    p01-introduction.main
2 items passed all tests:
   3 tests in p01-introduction.reverse_complement
   3 tests in p01-introduction.transcribe_dna
**********************************************************************
3 items had failures:
   4 of   4 in p01-introduction.count_character
   3 of   3 in p01-introduction.gc_content
   1 of   1 in p01-introduction.test_gc_content
14 tests in 7 items.
6 passed and 8 failed.
***Test Failed*** 8 failures.
```

Even though the script runs without any errors, doctest tells us that the functions are not working as the author intended!

Creating a doctest

Let's make a new function w ith doctest functionality:

- Start a new Python file called new Function.py
- Create a function that takes 2 inputs and multiplies them together
- Add a command to run the function, save the file and test it

```
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 thrown when running the commands
- Using the python interpreter answer the following 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.

Open time to work on Jupyter Notebook Biological Sequences or homework