Python Modules
Testing with Pytest
Unix commands

Learning to test functions

Announcements

- Midterm 1 is next week. To prepare, view:
  https://ucsb-cs8-f18.github.io/exam/e01/

- Gradescope: Submit your code as a pair. You must form a group on gradescope. Any one person in the group may submit

Different ways of using dbl

```
#doubling.py
def dbl( x ):
   return x*2
```

1. Call the function within the same file
2. Call the function from the Python shell
3. Call the function from another file < today>
4. Test the function with Pytest

Modules

Suppose we saved our dbl function in a file called doubling.py

We can now use the dbl function in another file:

```
#usingDoubling.py

import doubling #imports all the code in doubling
#from doubling import *  # Another way to import

print("Inside usingDoubling.py")
print(doubling.dbl(5))
print(doubling.dbl("UCSB"))
print(doubling.dbl([1, 7, 6, 5]))
```

This looks similar to our usage of the turtle module!

Conditional execution

Suppose that doubling.py included code that produced an output on the terminal

```
#doubling.py
def dbl( x ):
   return x*2

print(“Inside doubling.py”)
print(doubling.dbl(50))
```

What happens when usingDoubling.py is executed?

Conditional execution

Suppose that doubling.py included code that produced an output on the terminal

```
#doubling.py
def dbl( x ):
   return x*2

if __name__==“__main__”:
      print(“Inside doubling.py”)
      print(doubling.dbl(50))
```

Now the two print statements are only executed when we run doubling.py, not when we **import** it

Testing

```
#test_dbl.py
import pytest
from doubling import dbl

def test_dbl_1():
    assert dbl(0)==0

def test_dbl_2():
    assert dbl(2)==4

def test_dbl_3():
    assert dbl(“UCSB”)==“UCSBUCSB”
```

Run these tests from the unix command line:

```
$python3 –m pytest test_dbl.py
```

What is the output of this code

```
#doubling.py
def dbl( x ):
   return x*2

   print(doubling.dbl(50))
```

A. 100
B. 50
C. Error
D. No output

Using the math module

```
import math

print(math.sqrt(2))
print(math.sqrt(2)* math.sqrt(2) == 2)
```

What is printed by the last line?

A. True
B. False
C. Error

Demo

- In class we will code a few functions to understand floating point inaccuracies and working with pytest

- Write a function to find the absolute value of two numbers
- Write a function to find the distance between 2 points in 2D.

Unix commands

Learn how to navigate the unix file system using the following commands:

ls (list)

pwd (path to working directory)

cd (change directory)

mkdir (make a new directory)

Relative path vs. absolute path