CS150 - Docstrings & Unit Testing

Note Title 2/29/2012

Announcements

- HW5 is posted, due next Thursday
  - ↳ comments & documentation are required

Documentation

As we code larger projects, good documentation becomes absolutely essential.

Have you used help(——) yet?

Gives description of the behavior of functions, objects, etc.

This is different than comments (which are also essential) - comments are for coder, help is for user.

Docstrings

Python supports adding such documentation to any class.

(Note: This will be required on any hw.)

Standard:
- describe major purpose
- For functions, describe inputs and return values.

Syntax: """ info here """

three quotes

Example: Point class

add docstrings

Unit Testing

So far, to test our code we save it & reload Python for each test.

Repetitive! Have to reinitialize variables every time.

Faster: Write a script which loads your code & run it.

(or put it at bottom of py file)

Unit testing: another way

When running Python, the script which is started with the Python command is called __main__.

So

python testPoint.py ← __main__ (not MyPoint.py)

versus

python -i Point.py ← labeled __main__

vs.

python → then typing code

Using this fact:

__Example:__ file xxx.py

```
class XXX:
    #code for class

if __name__ == "__main__":
    #test code
```

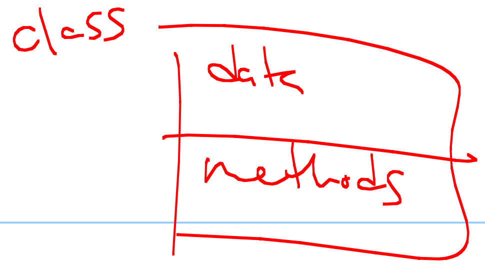
This test __only__ runs when we type python xxx.py (not when __module__ is imported)

Other Ways

- Python supports over 70 automated or semi-automated testing tools.

Oldest + most popular: PyUnit

- part of standard Python library
- write test separately (+ usually before coding the rest)



Theory of Unit testing

Unit testing is part of top-down design.
Idea is to carefully specify
requirements w/ behavior expected.
Then run tests often - at completion
or change of any included module,
at least.

Pros & Cons

Opponents of unit testing say it takes too much time and cost too much to set up & enforce.

Supporters say they can save much more by detecting errors early in the process.

How to use unittest (PyUnit)

Import it : import unittest

Unit teste themselves consist of functions the programmer writes to test the code.

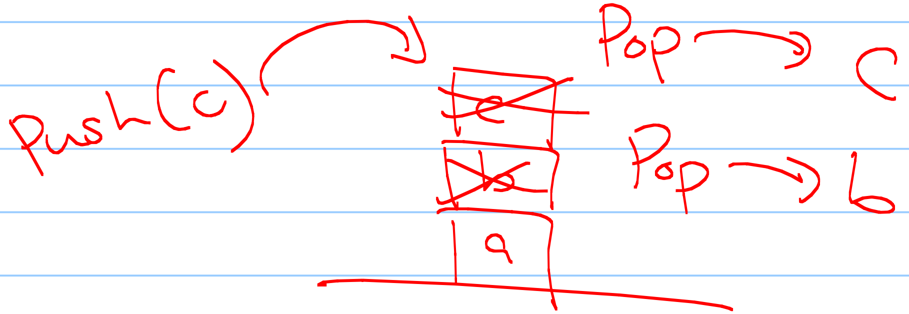
Eg:

```
if __name__ = "__main__":
    unittest.main()
```

Example: a stack

A simple data structure w/ 3 functions.

- Push(data)
- Pop()
- Peek()



(If empty, None is returned)

Let's code this, & then test it.