# Systems Programming in Python - Week 2

## Testing 1 of 2
## doctest & unittest

Brian Dorsey
brian@dorseys.org

# Announcements

0:45

# Lecture A (25)

0:20

# testing

lots

http://en.wikipedia.org/wiki/Software_testing

ignoring most of it today

module

class

function

lots of tools,
even at this level

doctest

unittest

pytest

nose

# demo time

# Test Driven Development?

how much testing should we do?

here's a starting point

- TDD if you can
- at least test the happy path
- add new tests before fixing each bug
  - prove that you understand the cause
  - make sure it never comes back

```
your code will have bugs, right?
```

# where to go from here

# mocks

# web testing

# Python Testing Tools Taxonomy

http://pycheesecake.org/wiki/PythonTestingToolsTaxonomy

# great intros:

http://www.doughellmann.com/PyMOTW/doctest/
http://www.doughellmann.com/PyMOTW/unittest/

# tools

http://garybernhardt.github.com/python-mock-comparison/
http://farmdev.com/thoughts/69/debugging-doctests-interactively/
http://sphinx.pocoo.org/ext/doctest.html
http://blog.ianbicking.org/javascript-doctest.html

# Lab

# Lab (optional)

- Test Driven Development practice

  Spend some time with the

  Python Koans (~20-40 minutes):

  **`http://python-koans.appspot.com/`**