

Think Python Chapter 16

Classes and Functions
David Rider

Code in GitHub

- `git clone https://github.com/riderd/think-python-16.git`

Time class

- put in time.py originally
 - my import of time.py interfered with built-in
 - moved to mytime.py

Pure functions

- does not modify any arguments to the function
- has no "side effects"
- just returns a value
- all versions of `add_time(t1, t2)` are pure
- `increment_pure_base60(time, seconds)` is a pure function
- functional programming style uses pure functions

Modifier Functions

- `increment(time, seconds)` is a modifier

Prototyping

- prototype and patch
 - aka code and fix
- mytime.py has several versions of functions that show evolution

Planning

- designed development (planning)
- Time object is a three digit number in base 60
- use integer arithmetic when everything converted to seconds
- make multiplication and subtraction much easier
- different methods in mytime.py using base60

Unit Testing

- enables 'refactoring'
- python unittest
 - based on JUnit framework
 - test case
 - use assertEquals, assertTrue, assertFalse, assertIsNot, assertRaises, ...
- helps alleviate some downsides of prototyping

Unit Testing

- showed problem in increment method when i misspelled attribute
 - `"seconds_time.seconds = seconds"`
- `setUp()` called before each test
 - note `self.t` vs. `t`
- `tearDown()` called after each test
- run with `python3 -m unittest test_mytime.py`
 - add `-v` for verbose
 - `test_time` changes `t` but only for that instance of test

Author code

- <http://thinkpython2.com/code/Time1.py>
- http://thinkpython2.com/code/Time1_soln.py

Exercise 1

Exercise 2

- write unit test for `day_from_date(d)`

Exercise 3

- Write unit test

Exercise 4

- I didn't do this and neither did author

Unit Testing Exercise

- write unit test that uses `assertRaises`
 - using `valid_time()` to check