5-minute intro to property-based testing in Python with `hypothesis`

Franklin Chen
http://FranklinChen.com/
http://ConscientiousProgrammer.com/

Pittsburgh Python Meetup
June 26, 2013

Property-based testing vs. example-based testing

Example-based testing: assert that a property is true for *some* values.

- Are list appends associative?
- Write some *examples*

```
def test_list_append_associative_example1():
    x, y, z = [2], [3, 4, 5], [6, 7]
    assert (x + y) + z == x + (y + z)

def test_list_append_associative_example2():
    x, y, z = [2, 3], [], [6, 7]
    assert (x + y) + z == x + (y + z)

# ...
```

Better: reduce boilerplate with parameters

pytest allows parameterization of arguments in tests.

- Rewrite multiple examples as a *single example*

```
import pytest

@pytest.mark.parametrize (("x", "y", "z"), [
    ([2], [3, 4, 5], [6, 7]),
    ([2, 3], [], [6, 7])
])
def test_list_append_associative_parametrized(x, y, z):
    assert (x + y) + z == x + (y + z)
```

Introduction to property-based hypothesis testing

```
from hypothesis.testdecorators import given

@given ([int], [int], [int])
def test_list_append_associative(x, y, z):
    assert (x + y) + z == x + (y + z)
```

- Passes!
- `hypothesis` *generates* a "large" number of examples
- Generation
  - Based on the *type* of the argument
  - Not exhaustive: failure to *falsify* does not mean true!
  - Default generators provided
  - You can write your own generators

A sample falsified hypothesis

```
@given (int, int)
def test_multiply_then_divide_is_same(x, y):
    assert (x * y) / y == x
```

Result:

```
... falsifying_example = ((0, 0), {})
```

Some links

- `hypothesis` documentation
- pytest-quickcheck
- My Pittsburgh Scala meetup talk on property-based testing using `ScalaCheck`
- Nat Pryce's blog post on June 23, 2013

Conclusion

- Property-based testing is a useful addition to your toolbox
- Try it out in Python!
- All materials for this talk available at https://github.com/franklinchen/lightning-talk-on-hypothesis
- Will write more about property-based testing on new blog http://ConscientiousProgrammer.com/