Hunting Bugs While Sleeping

Property-Based Testing with **Hypothesis**

Paul Amazona
Developer
@whatevergeek

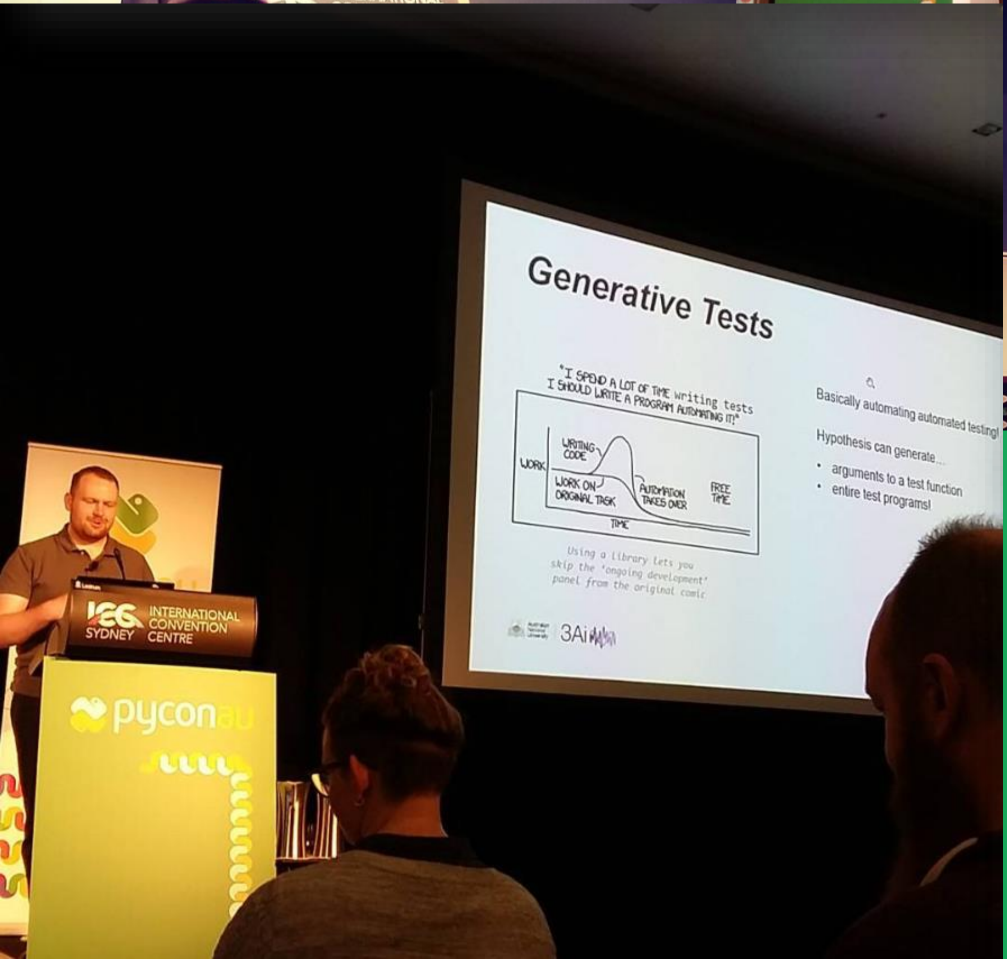
Django Against the Dark Arts
@attacus_au
INTERNATIONAL

PyCon AU 2018: Asyncio in (Micro)Python
ICC SYDNEY INTERNATIONAL CONVENTION CENTRE
pyconau

Container image security – Notary
Image repository.
Digitally signs images to be published.
The Update Framework (TUF) underlays the tool to maintain up-to-date software.
Allows user to verify content that they download.
Honourable mentions go to Clair
ICC SYDNEY INTERNATIONAL CONVENTION CENTRE
pyconau

Polylingualism
kboy3742
INTERNATIONAL CONVENTION CENTRE
pyconau

pyconau
iress
INTERNATIONAL CONVENTION


Generative Tests
"I SPEND A LOT OF TIME WRITING TESTS
I SHOULD WRITE A PROGRAM AUTOMATING IT!"
WRITING CODE
WORK
WORK ON ORIGINAL TASK
AUTOMATION TAKES OVER
FREE TIME
TIME
Using a library lets you skip the 'ongoing development' panel from the original comic
Basically automating automated testing!
Hypothesis can generate...
arguments to a test function
entire test programs!
3Ai
ICC SYDNEY INTERNATIONAL CONVENTION CENTRE
pyconau

pyconau

The Plan

- Hypothesis Overview
  - Property-Based Testing
  - Using Hypothesis
- Contributing to the Project
  - PyCon Sprints

Example Based Testing

```
class TestArithmetic(unittest.TestCase):
  def test_add(self):
    result = add(1, 2)
    self.assertEqual(3, result)
```

1. Set up some data.
2. Perform some operations on the data.
3. Assert something about the result.

Example Based Testing

```python
class TestArithmetic(unittest.TestCase):
  def test_add(self):
    result = add(1, 2)
    self.assertEqual(3, result)
```

```
Still rockin' @08:05:45 AM> python .\test_arithmetic.py
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
Still rockin' @08:08:25 AM>
```

Example Based Testing

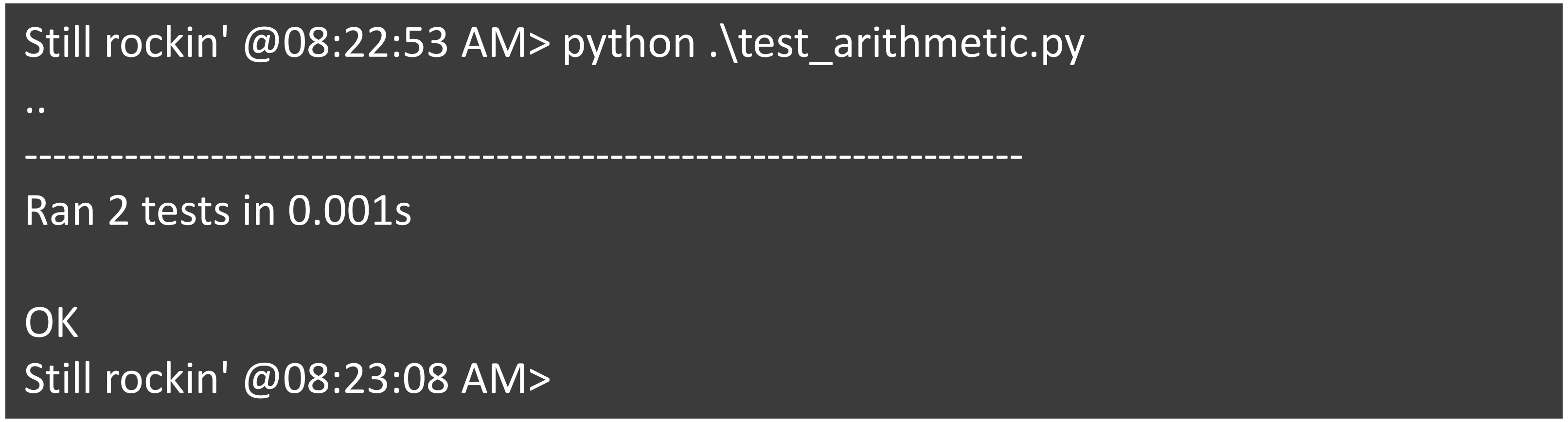
```python
def add(x, y):
  if (x==1 and y==2):
    return 3
  else:
    return 0
```

Example Based Testing

```
def test_add(self):
    result = add(1, 2)
    self.assertEqual(3, result)

def test_add2(self):
    result = add(2, 2)
    self.assertEqual(4, result)
```

Example Based Testing

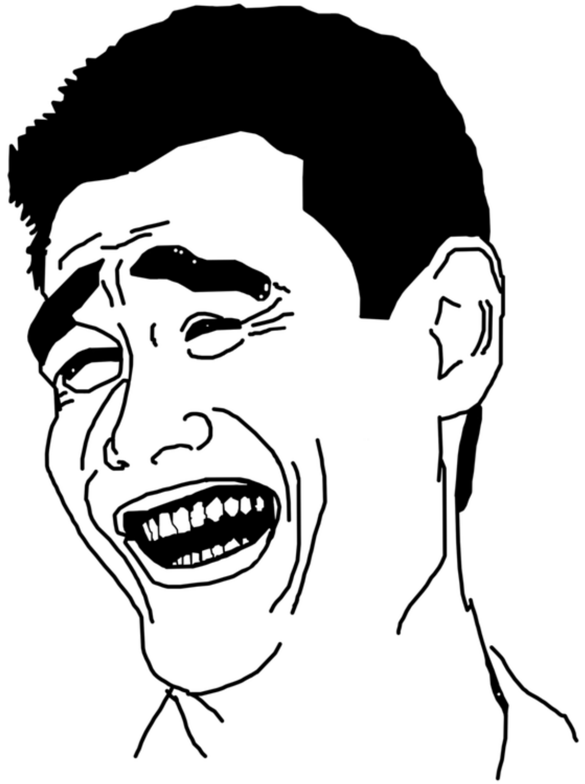
```python
def test_add(self):
    result = add(1, 2)
    self.assertEqual(3, result)

def test_add2(self):
    result = add(2, 2)
    self.assertEqual(4, result)
```



```
Still rockin' @08:22:53 AM> python .\test_arithmetic.py
..
----------------------------------------------------------------------
Ran 2 tests in 0.001s

OK
Still rockin' @08:23:08 AM>
```

Example Based Testing

```
def add(x, y):
    if (x==1 and y==2):
        return 3
    elif (x==2 and y==2):
        return 4
    else:
        return 0
```



Example Based Testing

```
def test_add(self):
    x = random.randint(0,1000)
    y = random.randint(0,1000)
    result = add(x, y)
    self.assertEqual(x + y, result)
```

Example Based Testing

```python
def test_add(self):
    x = random.randint(0,1000)
    y = random.randint(0,1000)
    result = add(x, y)
    self.assertEqual(x + y, result)
```

```
FAIL: test_add (__main__.TestArithmetic)
----------------------------------------------------------------------
Traceback (most recent call last):
  File ".\test_arithmetic.py", line 18, in test_add
    self.assertEqual(x + y, result)
AssertionError: 86 != 0


----------------------------------------------------------------------
Ran 3 tests in 0.001s

FAILED (failures=1)
```

Example Based Testing

```
def test_add(self):
  for i in range(0, 100):
    x = random.randint(0,1000)
    y = random.randint(0,1000)
    result = add(x, y)
    self.assertEqual(x + y, result)
```

Example Based Testing

```
def test_add(self):
  for i in range(0, 100):
    x = random.randint(0,1000)
    y = random.randint(0,1000)
    result = add(x, y)
    self.assertEqual(x + y, result)
```

Implementation duplication

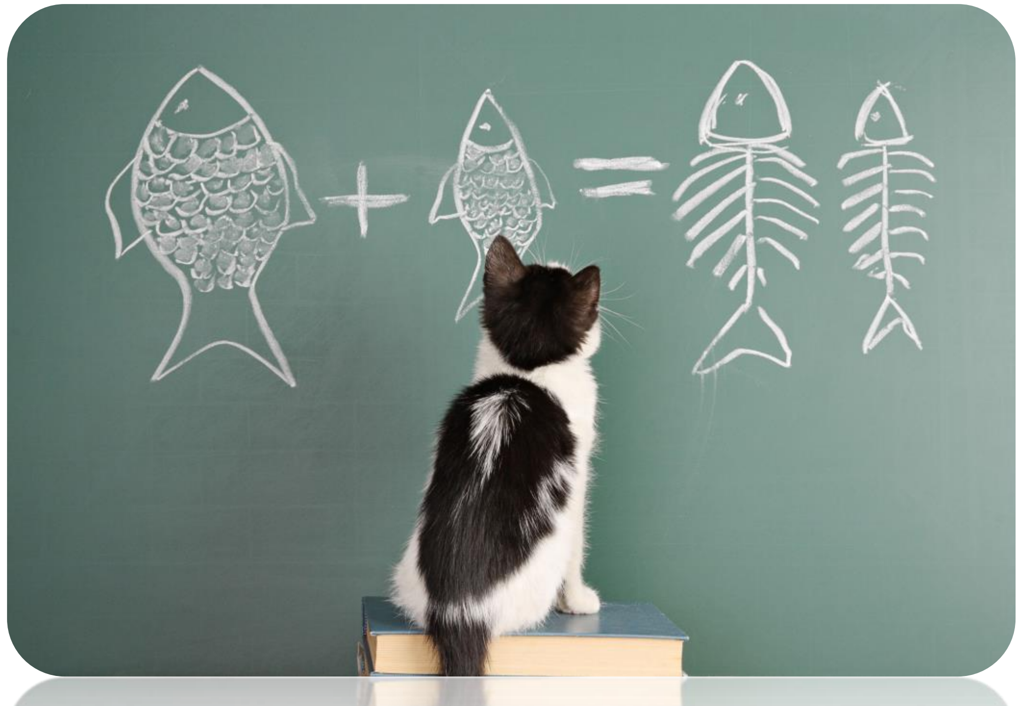
Example Based Testing

```python
def test_add(self):
  for i in range(0, 100):
    x = random.randint(0,1000)
    y = random.randint(0,1000)
    result = add(x, y)
    self.assertEqual(x + y, result)
```

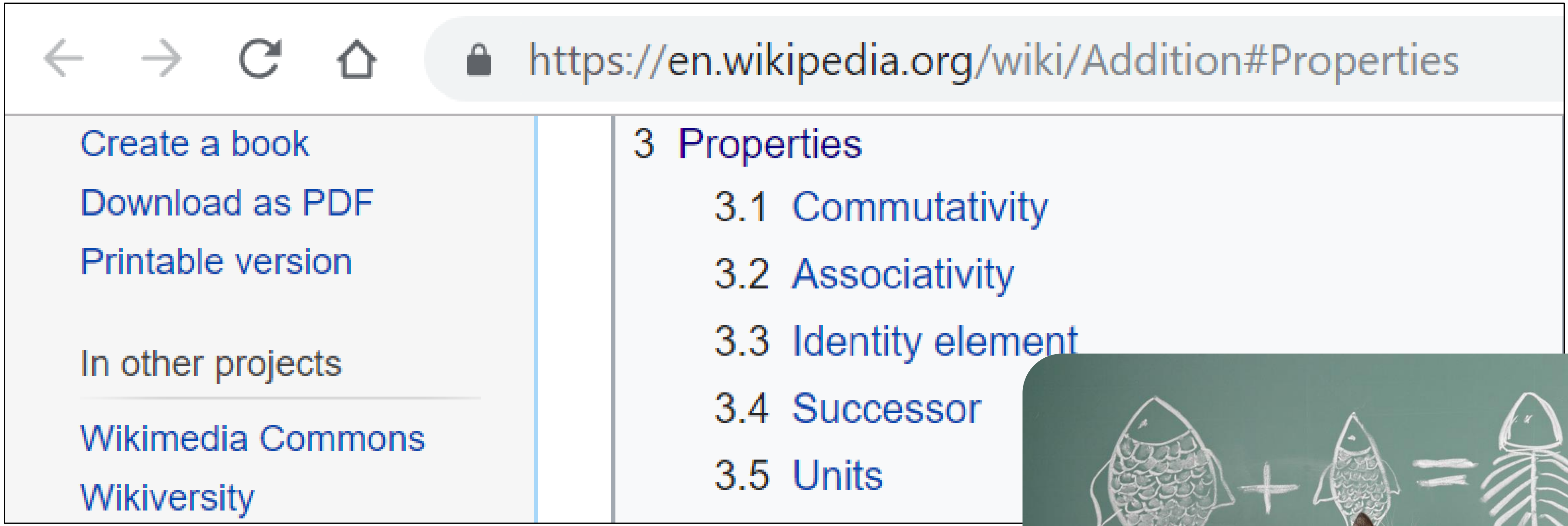
Implementation duplication

But, how else can you test?

What is Addition?



What is Addition?



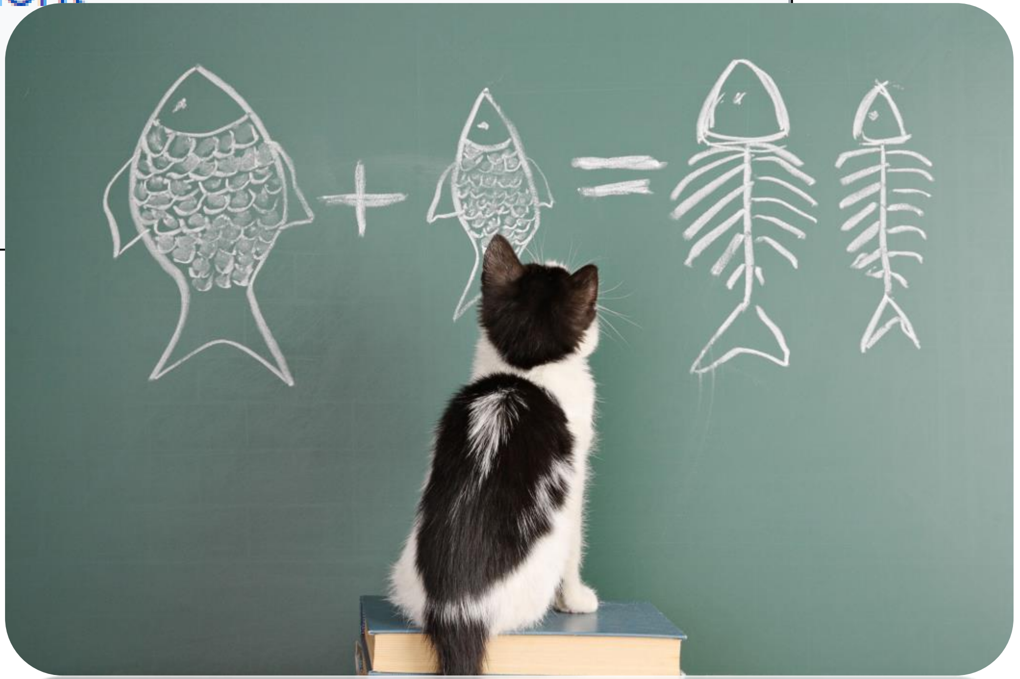
https://en.wikipedia.org/wiki/Addition#Properties

Create a book
Download as PDF
Printable version

In other projects

Wikimedia Commons
Wikiversity

3 Properties
- 3.1 Commutativity
- 3.2 Associativity
- 3.3 Identity element
- 3.4 Successor
- 3.5 Units



Property-Based Testing

```python
from arithmetic import *
from hypothesis import given, strategies as st

@given(st.integers(), st.integers())
def test_commutative_property(x, y):
    assert add(x,y) == add(y,x)

@given(st.integers())
def test_add1twice_is_add2once_property(x):
    assert add(add(x,1),1) == add(x,2)

@given(st.integers())
def test_identity_property(x):
    assert add(x, 0) == x
```

1. For all data matching some specification.
2. Perform some operations on the data.
3. Assert something about the result.

Hypothesis is a library designed to help you write what are called *property-based tests*.

Property-Based Testing

1. For all data matching some specification.
2. Perform some operations on the data.
3. Assert something about the result.

```python
from arithmetic import *
from hypothesis import given, strategies as st

@given(st.integers(), st.integers())
def test_commutative_property(self, x, y):
    assert add(x,y) == add(y,x)

@given(st.integers())
def test_add1twice_is_add2once_property(self, x):
    assert add(add(x,1),1) == add(x,2)

@given(st.integers())
def test_identity_property(self, x):
    assert add(x, 0) == x
```

```
Still rockin' @09:05:16 AM> pytest .\test_arithmetic_properties.py --hypothesis-show-statistics
============================ test session starts ============================
platform win32 -- Python 3.6.6, pytest-3.7.3, py-1.5.4, pluggy-0.7.1
rootdir: D:\Laboratory\Sandbox\PythonStuff\hypothesis_stuff\pugs_demo\rosetta, inifile:
plugins: xdist-1.23.0, forked-0.2, flaky-3.4.0, hypothesis-3.69.2
collected 3 items

test_arithmetic_properties.py ...                                   [100%]
============================ Hypothesis Statistics ============================

test_arithmetic_properties.py::TestArithmetic::test_add1twice_is_add2once_property:

  - 100 passing examples, 0 failing examples, 0 invalid examples
  - Typical runtimes: 0-15 ms
  - Fraction of time spent in data generation: ~ 66%
  - Stopped because settings.max_examples=100

test_arithmetic_properties.py::TestArithmetic::test_commutative_property:
  - 100 passing examples, 0 failing examples, 0 invalid examples
  - Typical runtimes: 0-15 ms
  - Fraction of time spent in data generation: ~ 23%
  - Stopped because settings.max_examples=100

test_arithmetic_properties.py::TestArithmetic::test_identity_property:
  - 100 passing examples, 0 failing examples, 0 invalid examples
  - Typical runtimes: 0-15 ms
  - Fraction of time spent in data generation: ~ 15%
  - Stopped because settings.max_examples=100


===Flaky Test Report===
===End Flaky Test Report===


========================= 3 passed in 1.00 seconds ============================
```

Property-Based Testing

```python
def add(x, y):
    if (x==1 and y==2):
        return 3
    elif (x==2 and y==2):
        return 4
    else:
        return 0
```

```
…

self = <test_arithmetic_properties.TestArithmetic
testMethod=test_add1twice_is_add2once_property>
x = 1

    @given(st.integers())
    def test_add1twice_is_add2once_property(self, x):
>       assert add(add(x,1),1) == add(x,2)
E       AssertionError: assert 0 == 3
E        +  where 0 = add(0, 1)
E        +    where 0 = add(1, 1)
E        +  and   3 = add(1, 2)
…
```

What can you generate?

integers, booleans, floats, tuples, sample_from, lists, sets, iterables, dictionaries, streaming, characters, text, binary, fractions, decimals, permutations, datetimes, etc… etc… etc

https://tinyurl.com/hypo-data

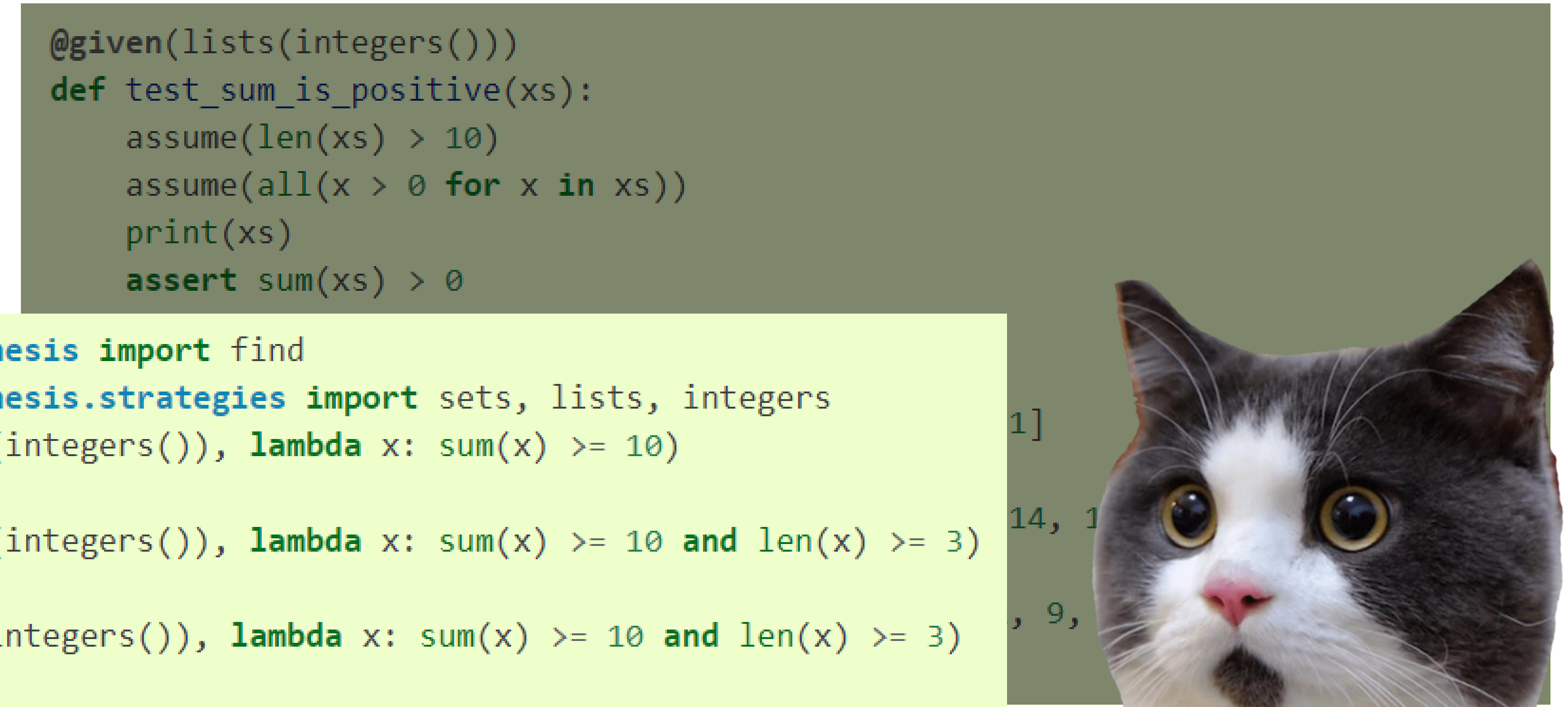
Other notable features

- Example Database
- Assumptions

```
@given(lists(integers()))
def test_sum_is_positive(xs):
    assume(len(xs) > 10)
    assume(all(x > 0 for x in xs))
    print(xs)
    assert sum(xs) > 0

In: test_sum_is_positive()
[17, 12, 7, 13, 11, 3, 6, 9, 8, 11, 47, 27, 1, 31, 1]
[6, 2, 29, 30, 25, 34, 19, 15, 50, 16, 10, 3, 16]
[25, 17, 9, 19, 15, 2, 2, 4, 22, 10, 10, 27, 3, 1, 14, 1
[17, 65, 78, 1, 8, 29, 2, 79, 28, 18, 39]
[13, 26, 8, 3, 4, 76, 6, 14, 20, 27, 21, 32, 14, 42, 9,
[2, 1, 2, 2, 3, 10, 12, 11, 21, 11, 1, 16]
```

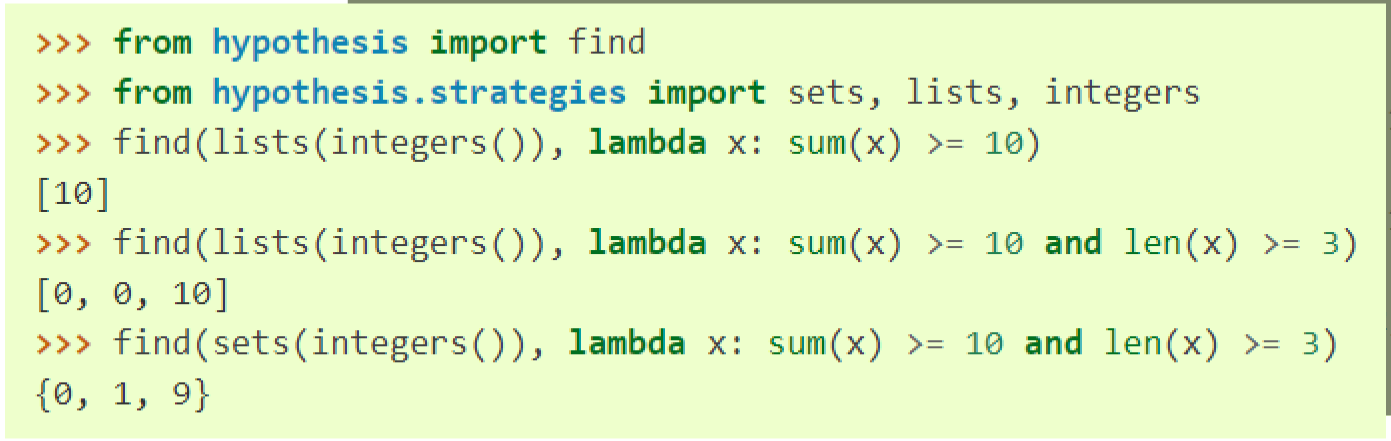
Other notable features

- Example Database
- Assumptions



```
@given(lists(integers()))
def test_sum_is_positive(xs):
    assume(len(xs) > 10)
    assume(all(x > 0 for x in xs))
    print(xs)
    assert sum(xs) > 0
```

```
1]
14, 1
, 9,
```



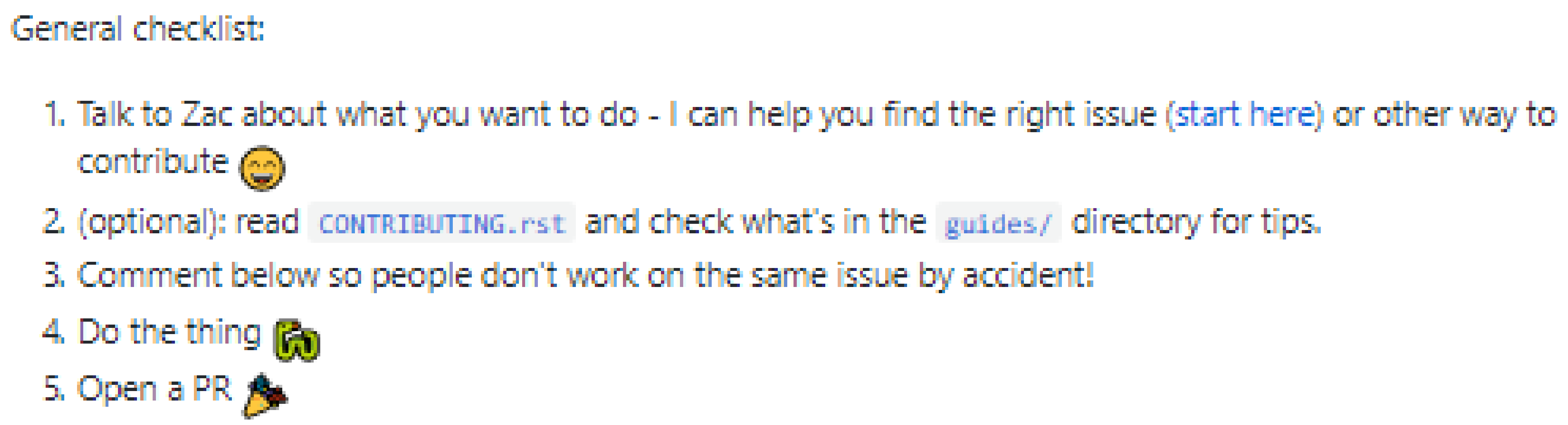
```
>>> from hypothesis import find
>>> from hypothesis.strategies import sets, lists, integers
>>> find(lists(integers()), lambda x: sum(x) >= 10)
[10]
>>> find(lists(integers()), lambda x: sum(x) >= 10 and len(x) >= 3)
[0, 0, 10]
>>> find(sets(integers()), lambda x: sum(x) >= 10 and len(x) >= 3)
{0, 1, 9}
```

- Finding Values

PyCon Sprints

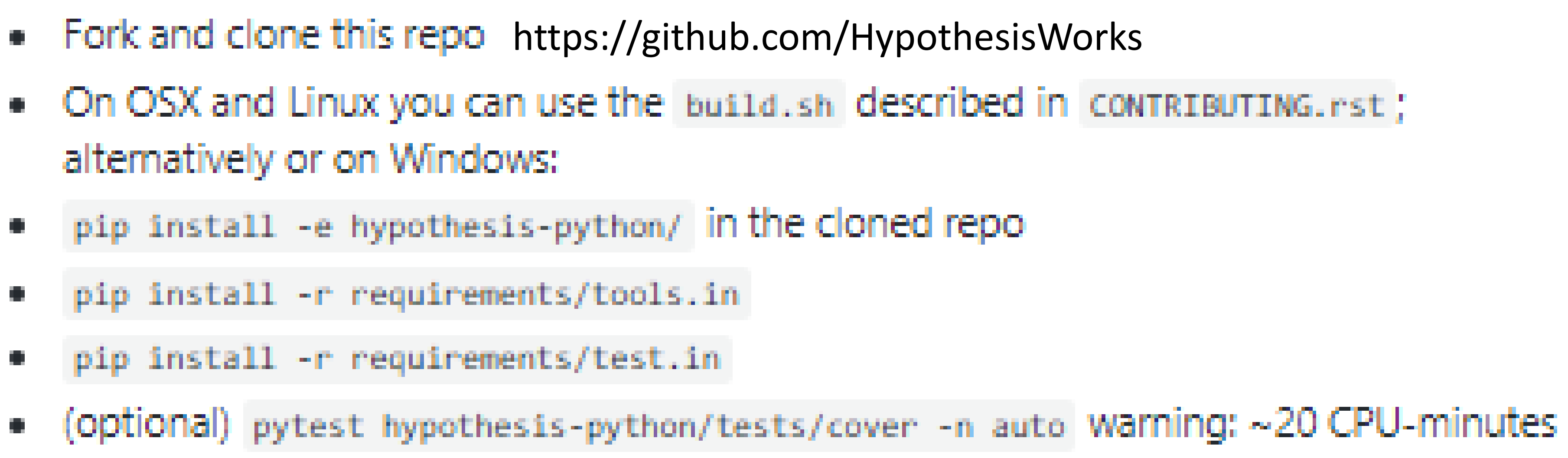
- Fix a bug
  - Issue: https://github.com/HypothesisWorks/hypothesis/issues/1317
  - Solution: https://github.com/HypothesisWorks/hypothesis/pull/1515
- Test some code (Debugging)
  - https://github.com/HypothesisWorks/hypothesis/issues/1493

PyCon Sprints



General checklist:

1. Talk to Zac about what you want to do - I can help you find the right issue (start here) or other way to contribute 😄
2. (optional): read `CONTRIBUTING.rst` and check what's in the `guides/` directory for tips.
3. Comment below so people don't work on the same issue by accident!
4. Do the thing
5. Open a PR 🎉

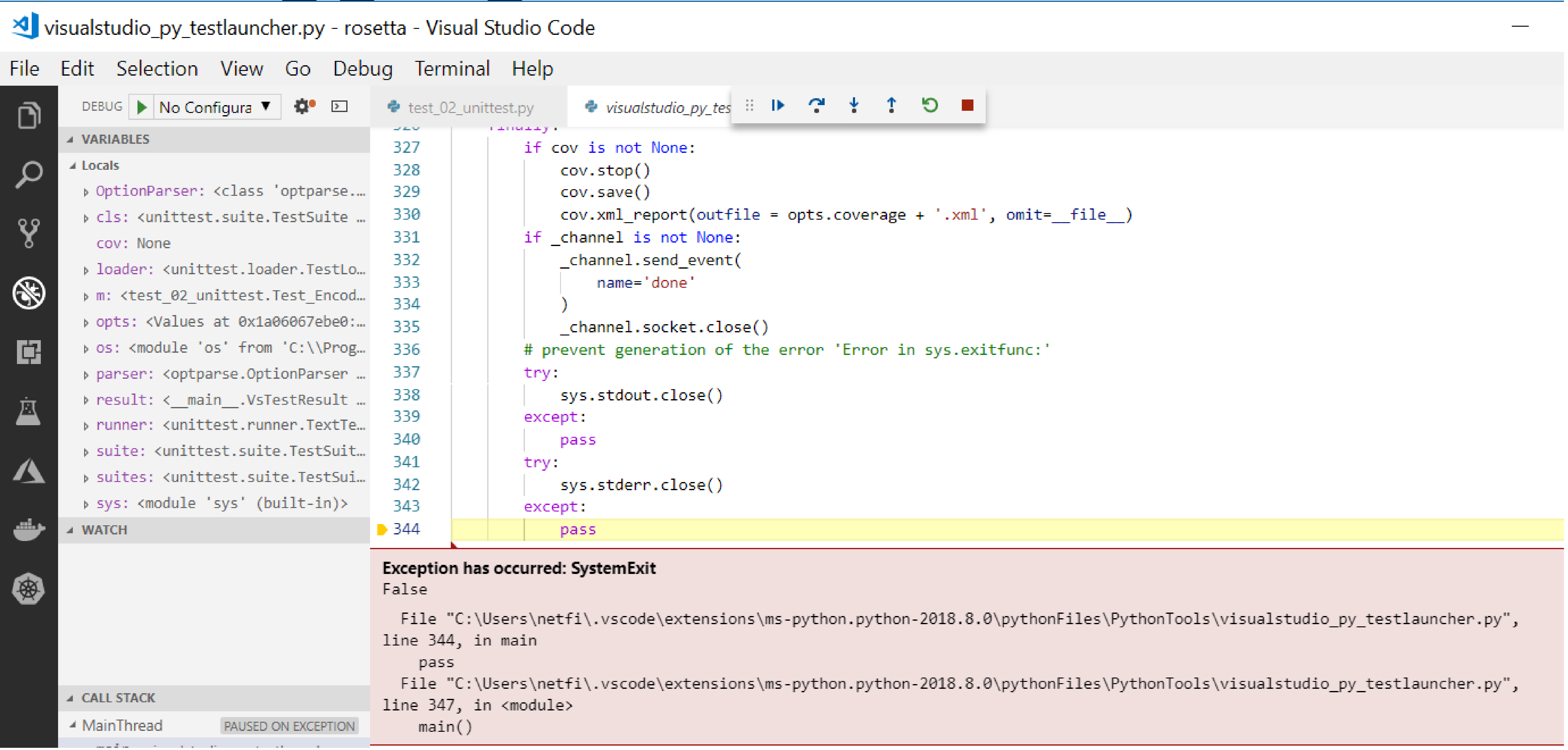


- Fork and clone this repo https://github.com/HypothesisWorks
- On OSX and Linux you can use the `build.sh` described in `CONTRIBUTING.rst`; alternatively or on Windows:
- `pip install -e hypothesis-python/` in the cloned repo
- `pip install -r requirements/tools.in`
- `pip install -r requirements/test.in`
- (optional) `pytest hypothesis-python/tests/cover -n auto` warning: ~20 CPU-minutes

Debugging

```
from run_length_encoding import *

from hypothesis import given, example
from hypothesis.strategies import text

import unittest

class Test_Encoding(unittest.TestCase):
@given(text())
def test_decode_inverts_encode(self, s):
    self.assertEqual(decode(encode(s)), s)

if __name__ == '__main__’:
    unittest.main()
```

Debugging



```
OptionParser: <class 'optparse....
cls: <unittest.suite.TestSuite ...
cov: None
loader: <unittest.loader.TestLo...
m: <test_02_unittest.Test_Encod...
opts: <Values at 0x1a06067ebe0:...
os: <module 'os' from 'C:\\Prog...
parser: <optparse.OptionParser ...
result: <__main__.VsTestResult ...
runner: <unittest.runner.TextTe...
suite: <unittest.suite.TestSuit...
suites: <unittest.suite.TestSui...
sys: <module 'sys' (built-in)>
```

```python
        if cov is not None:
            cov.stop()
            cov.save()
            cov.xml_report(outfile = opts.coverage + '.xml', omit=__file__)
        if _channel is not None:
            _channel.send_event(
                name='done'
            )
            _channel.socket.close()
        # prevent generation of the error 'Error in sys.exitfunc:'
        try:
            sys.stdout.close()
        except:
            pass
        try:
            sys.stderr.close()
        except:
            pass
```

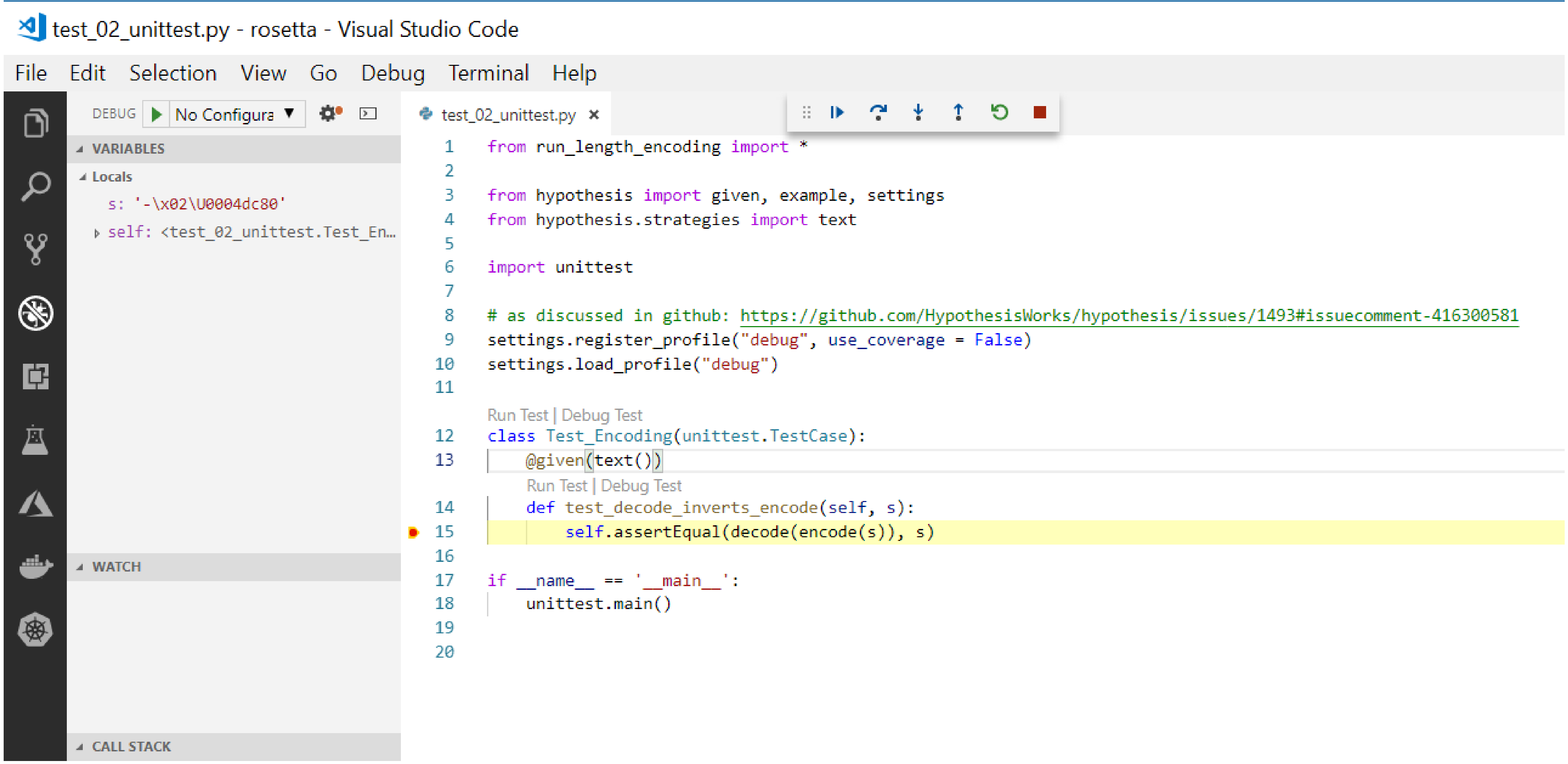
Exception has occurred: SystemExit

```
False

  File "C:\Users\netfi\.vscode\extensions\ms-python.python-2018.8.0\pythonFiles\PythonTools\visualstudio_py_testlauncher.py", line 344, in main
    pass
  File "C:\Users\netfi\.vscode\extensions\ms-python.python-2018.8.0\pythonFiles\PythonTools\visualstudio_py_testlauncher.py", line 347, in <module>
    main()
```

Debugging

```
from run_length_encoding import *

from hypothesis import given, example, settings
from hypothesis.strategies import text

import unittest

# as discussed in github: https://github.com/HypothesisWorks/hypothesis/issues/1493#issuecomment-416300581
settings.register_profile("debug", use_coverage = False)
settings.load_profile("debug")

class Test_Encoding(unittest.TestCase):
@given(text())
def test_decode_inverts_encode(self, s):
    self.assertEqual(decode(encode(s)), s)

if __name__ == '__main__’:
    unittest.main()
```

Debugging


test_02_unittest.py - rosetta - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
DEBUG
No Configura
VARIABLES
Locals
WATCH
CALL STACK
test_02_unittest.py
Run Test | Debug Test

```
s: '-\x02\U0004dc80'
self: <test_02_unittest.Test_En…
```

```python
from run_length_encoding import *

from hypothesis import given, example, settings
from hypothesis.strategies import text

import unittest

# as discussed in github: https://github.com/HypothesisWorks/hypothesis/issues/1493#issuecomment-416300581
settings.register_profile("debug", use_coverage = False)
settings.load_profile("debug")

class Test_Encoding(unittest.TestCase):
    @given(text())
    def test_decode_inverts_encode(self, s):
        self.assertEqual(decode(encode(s)), s)

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

Resources

- https://hypothesis.works
- https://hypothesis.readthedocs.io

"The larger purpose of Hypothesis is to drag the world kicking and screaming into a new and terrifying age of high quality software."
- David R. MacIver