Teaching test-driven development with pytest

Ken Youens-Clark

Remote Python Pizza

25 April 2020

https://github.com/kyclark/remote.python.pizza

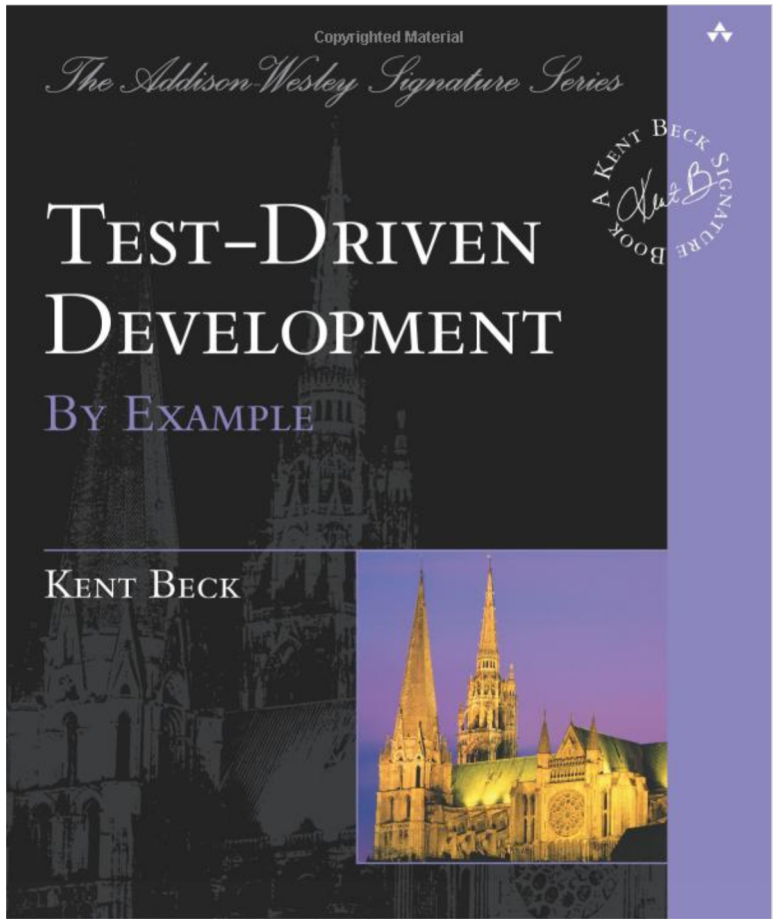
Write a Python program called add.py that takes two integer values provided as positional command-line arguments and shows the result of adding them:

```
$ ./add.py 1 2
1 + 2 = 3
```

Test-Driven Development

1. Add a test
2. Run all tests and see if the new test fails
3. Write the code
4. Run tests
5. Refactor code
6. Repeat

https://en.wikipedia.org/wiki/Test-driven_development



Specifications

- The program accepts only two valid integer values.
- The program prints helpful "usage" statements upon request and errors.
- The program correctly adds the two values and displays the sum.

The program accepts only two valid integer values

```
$ ./add.py 1
usage: add.py [-h] int int
add.py: error: the following arguments are required: int

$ ./add.py 1 foo
usage: add.py [-h] int int
add.py: error: argument int: invalid int value: 'foo'
```

Usage statements upon request and errors

```
$ ./add.py -h
usage: add.py [-h] int int

Add two integer values

positional arguments:
  int         Two numbers to add

optional arguments:
  -h, --help  show this help message and exit
```

Correctly adds the two values

```
$ ./add.py 1 2
1 + 2 = 3

$ ./add.py 3 8
3 + 8 = 11
```

Testing

- There is a file called “add.py” in the current directory
- The program is runnable as “./add.py”
- The program will print “usage” for “-h” and --help”
- The program will show errors for any number of arguments not 2
- The program will reject arguments that are not integers
- The program will properly add two numbers

Almost correct

```
def main():
    args = get_args()
    n1, n2 = args.numbers
    print(n1 + n2)
```

```
$ ./add.py 1 2
3
```

```
$ make test
pytest -xv test.py
============================= test session starts ==============================
platform darwin -- Python 3.8.1, pytest-5.3.5, py-1.8.1, pluggy-0.13.1 --
/Library/Frameworks/Python.framework/Versions/3.8/bin/python3
cachedir: .pytest_cache
rootdir: /Users/kyclark, inifile: pytest.ini
collected 5 items

test.py::test_exists PASSED                                              [ 20%]
test.py::test_usage PASSED                                               [ 40%]
test.py::test_wrong_number_args PASSED                                   [ 60%]
test.py::test_not_numbers PASSED                                         [ 80%]
test.py::test_valid_input FAILED                                         [100%]

=================================== FAILURES ===================================
_____________________________ test_valid_input _____________________________

    def test_valid_input():
        """test with valid input"""

        for x, y, z in [[0, 0, 0], [1, 0, 1], [1, 2, 3], [2, 1, 3]]:
            rv, out = getstatusoutput(f'{prg} {x} {y}')
            assert rv == 0
>           assert out.rstrip() == f'{x} + {y} = {z}'
E           AssertionError: assert '0' == '0 + 0 = 0'
E             - 0
E             + 0 + 0 = 0

test.py:63: AssertionError
!!!!!!!!!!!!!!!!!!!!!! stopping after 1 failures !!!!!!!!!!!!!!!!!!!!!!
========================= 1 failed, 4 passed in 0.41s ==========================
make: *** [test] Error 1
```

Reading test failures

```
E               AssertionError: assert '0' == '0 + 0 = 0'
E                 - 0
E                 + 0 + 0 = 0
```

Corrected

```
def main():
    args = get_args()
    n1, n2 = args.numbers
    print(f'{n1} + {n2} = {n1 + n2}')
```

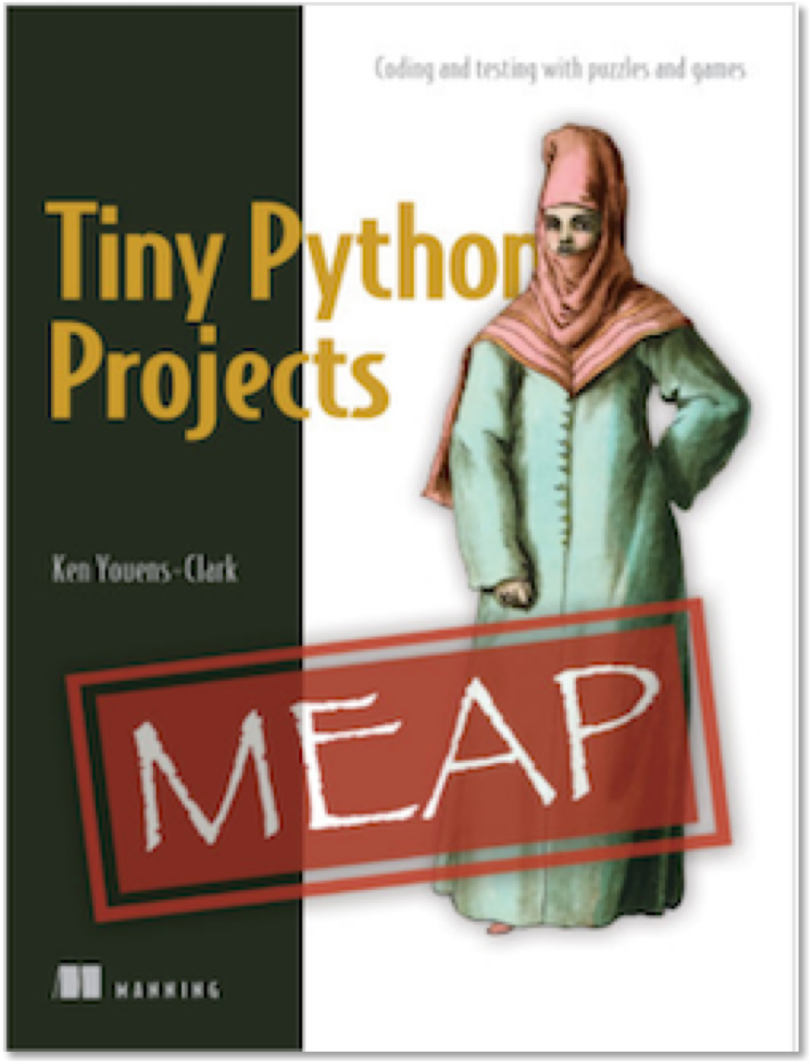
```
$ ./solution.py 1 2
1 + 2 = 3
```

Encoding and running test with pytest

```
$ make test
pytest -xv test.py
============================= test session starts ==============================
...

test.py::test_exists PASSED                                              [ 20%]
test.py::test_usage PASSED                                               [ 40%]
test.py::test_wrong_number_args PASSED                                   [ 60%]
test.py::test_not_numbers PASSED                                         [ 80%]
test.py::test_valid_input PASSED                                         [100%]

============================== 5 passed in 0.52s ===============================
```



Presentation and code:
https://github.com/kyclark/remote.python.pizza

Tiny Python Projects at Manning:
https://www.manning.com/books/tiny-python-projects

GitHub repo:
https://github.com/kyclark/tiny_python_projects

YouTube videos:
https://www.youtube.com/user/kyclark