



ISMRM 27TH ANNUAL MEETING & EXHIBITION

Palais des congrès de Montréal  Montréal, QC, Canada  11–16 May 2019

Secret Session: Master coding in your research environment

Testing your code

Mathieu Boudreau, PhD

Research Fellow, Montreal Heart Institute

Software Developer for qMRLab & AxonDeepSeg @ NeuroPoly, Polytechnique de Montréal



INSTITUT DE
CARDIOLOGIE
DE MONTRÉAL



AxonDeepSeg



POLYTECHNIQUE
MONTRÉAL



What are tests?

A way to reproduce and automate an evaluation of the expected behaviour of your code.

Tests will...

- inform you of where your code is behaving unexpectedly
- describe expected behaviour and the actual behaviour

Broad types of tests

Unit tests

- For individual pieces of code (e.g. one function/file in isolation)

Integration Tests

- Interaction between multiple functions/files (i.e. a user experience)

How do I start?

Most coding languages have testing frameworks!

MATLAB → *MATLAB Unit Test Framework* and *MOxUnit*

Python → *PyTest* and *unittest*

Octave → *MOxUnit*

R → *testthat* and *RUnit*

C++ → *Catch2* and *Googletest*

C → *Check* and *Unity*

... and many more!

Simple example

Example function

```
add.py  X  
You, 3 days ago | 1 author (You)  
1  
2  
3 def add(a, b):  
4     """ Add two mathematical objects together  
5     :return: sum of a and b  
6     """  
7  
8     return a + b  
9
```

Simple example

Example test

test_add.py

```
You, 3 days ago | 1 author (You)
1 # coding: utf-8
2
3 import pytest
4 from testing_your_code.math.add import add
5
6
You, 3 days ago | 1 author (You)
7 class TestCore(object):
8
9     # -----add.py tests----- #
10    def test_add_known_integer_case(self):
11        a = 2
12        b = 2
13
14        expected_value = 4
15        actual_value = add(a,b)
16
17        assert actual_value == expected_value
18
```

Simple example

Run the tests!

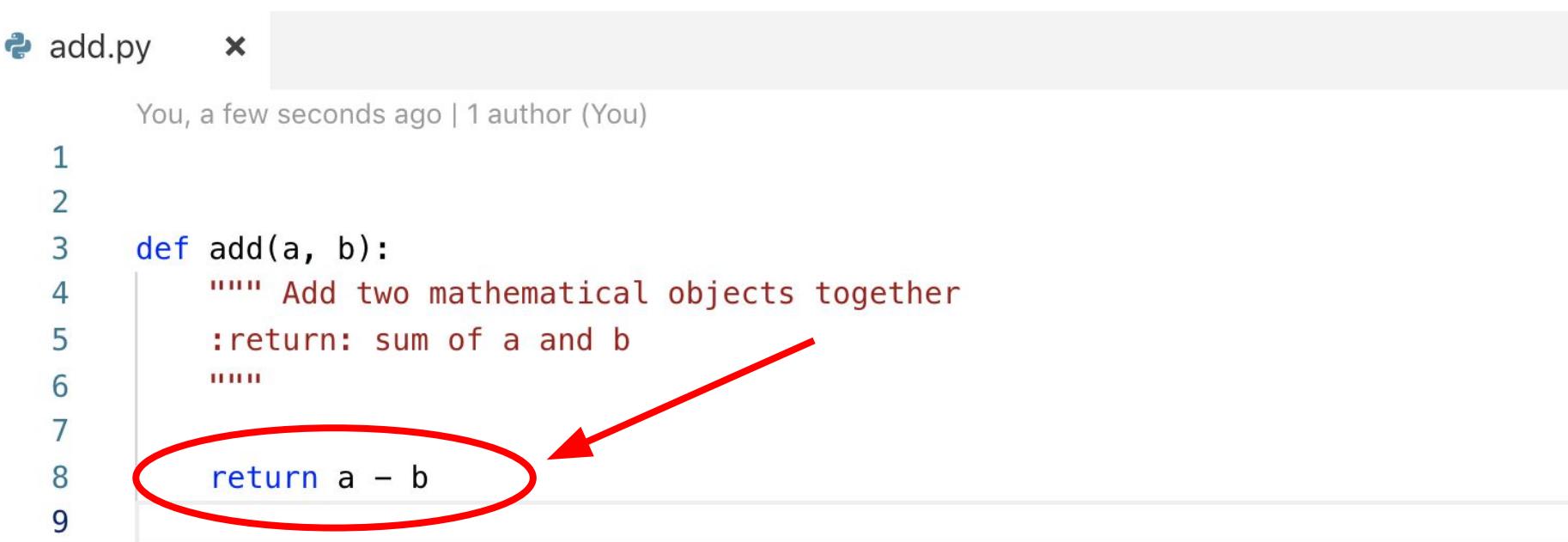
```
[base) mathieuboudreau@brassens:~/neuropoly/github/testing-your-code$ py.test
=====
===== test session starts =====
platform darwin -- Python 3.7.1, pytest-3.6.0, py-1.7.0, pluggy-0.6.0
rootdir: /Users/mathieuboudreau/neuropoly/github/testing-your-code, infile:
plugins: remotedata-0.3.1, openfiles-0.3.1, doctestplus-0.2.0, cov-2.5.1, arraydiff-0.3
collected 1 item

tests/math/test_add.py .

=====
1 passed in 0.04 seconds =====
[100%]
```

Simple example

Without realizing, while developing a “subtract” function, you accidentally save this...



add.py

You, a few seconds ago | 1 author (You)

```
1
2
3 def add(a, b):
4     """ Add two mathematical objects together
5     :return: sum of a and b
6     """
7
8     return a - b
9
```

Simple example

Run the test again, and...

```
(base) mathieuboudreau@brassens:~/neuropoly/github/testing-your-code$ py.test
=====
platform darwin -- Python 3.7.1, pytest-3.6.0, py-1.7.0, pluggy-0.6.0
rootdir: /Users/mathieuboudreau/neuropoly/github/testing-your-code, inifile:
plugins: remotedata-0.3.1, openfiles-0.3.1, doctestplus-0.2.0, cov-2.5.1, arraydiff-0.3
collected 1 item

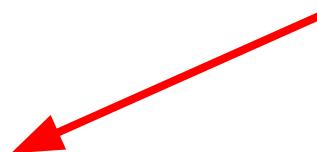
tests/math/test_add.py F [100%]

=====
FAILURES =====
----- TestCore.test_add_known_integer_case -----
self = <tests.math.test_add.TestCore object at 0x109377b38>

    def test_add_known_integer_case(self):
        a = 2
        b = 2

        expected_value = 4
        actual_value = add(a,b)
>       assert actual_value == expected_value
E       assert 0 == 4

tests/math/test_add.py:17: AssertionError
=====
1 failed in 0.09 seconds =====
```



Real-world example

2 year old PhD code: <https://github.com/mathieuboudreau/qmt-optimization>

Updated a software dependency (qMRLab)

Failure Summary:

| Name | Failed | Incomplete | Reason(s) |
|--|--------|------------|-----------|
| <hr/> | | | |
| SPGR_MonteCarlo_Test/test_SPGR_MonteCarlo_fit_returns_expected_sizes | X | X | Errored. |
| <hr/> | | | |
| generateSPGRSimParam_Test/testgenerateSPGRSimParam_generates_demo_values | X | X | Errored. |

result =

1x68 [TestResult](#) array with properties:

Name
Passed
Failed
Incomplete
Duration
Details

Totals:
66 Passed, 2 Failed ([rerun](#)), 2 Incomplete.
1167.0458 seconds testing time.



>>

Real-world example

Where and why did they fail?

Running generateSPGRSimParam_Test

```
=====
Error occurred while setting up or tearing down generateSPGRSimParam_Test.
As a result, all generateSPGRSimParam_Test tests failed and did not run to completion.
```

Error ID:

'MATLAB:load:couldNotReadFile'

Error Details:

Error using [load](#)

Unable to read file

'.../.../.../qMRLab/Models_Functions/SPGRfun/Parameters/DefaultSim.mat'. No such file
or directory.

Error in [generateSPGRSimParam_Test/getDefaultSPGRParams](#) (line 18)

```
load('.../.../.../qMRLab/Models_Functions/SPGRfun/Parameters/DefaultSim.mat')
```

=====

Done generateSPGRSimParam_Test

Real-world example

Where and why did they fail?

```
=====
Error occurred in SPGR_MonteCarlo_Test/test_SPGR_MonteCarlo_fit_returns_expected_sizes and it did not run to completion

-----
Error ID:
-----
'MATLAB:TooManyInputs'

-----
Error Details:
-----
Error using FitData
Too many input arguments.

-----
Error in SPGR MonteCarlo/fit (line 13)
    fitResults.noiselessDataset = FitData(dataStruct, protocolStruct,
    fitOptStruct, 'SPGR', 0);

Error in SPGR MonteCarlo Test/test\_SPGR MonteCarlo fit returns expected sizes
(line 251)
    MCobj.fit();
=====
```

The image shows a screenshot of a MATLAB error message. Several parts of the message are highlighted with red annotations: the error ID 'MATLAB:TooManyInputs' is circled with a red oval and has a red arrow pointing to it from the top left; the entire line 'Error in SPGR MonteCarlo/fit (line 13)' is circled with a red oval and has a red arrow pointing to it from the bottom left.

If you there's a possibility that you encounter one of these situations, tests are for you!

- A software, library, or operating system upgrade
 - MATLAB 2015a → MATLAB 2019a, numpy 1.8.0 → 1.16.2
- A change in operating system or hardware (new computer)
 - MacOS Mojave → Windows 10
- You have a bad memory
 - How do I use this function again...?
- Plan on using your code in the future
 - (in the year 2023) Why won't this work anymore..?
- You make mistakes writing code
 - i.e. everyone should write tests!

Awesome features that having tests open up

Continuous Integration

- Automatically runs your tests when you commit to GitHub



Travis CI



Jenkins

Code Coverage

- Informs you of which code lines your tests are evaluating, and those that missed.



COVERALLS



(CodeCov)

Example repositories containing tests

Simple examples

- <https://github.com/ismrm-coding-secret-session/testing-your-code>

Real-world examples

- Python: <https://github.com/neuropoly/axondeepseq>
- Octave/MATLAB: <https://github.com/qmrlab/qmrlab>
- MATLAB: <https://github.com/mathieuboudreau/qmt-optimization>