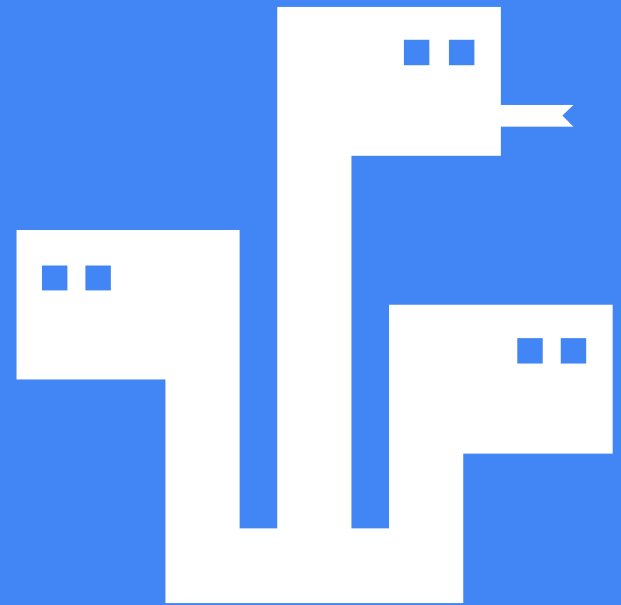


Testing

Python - Nick Reynolds
May 5, 2017



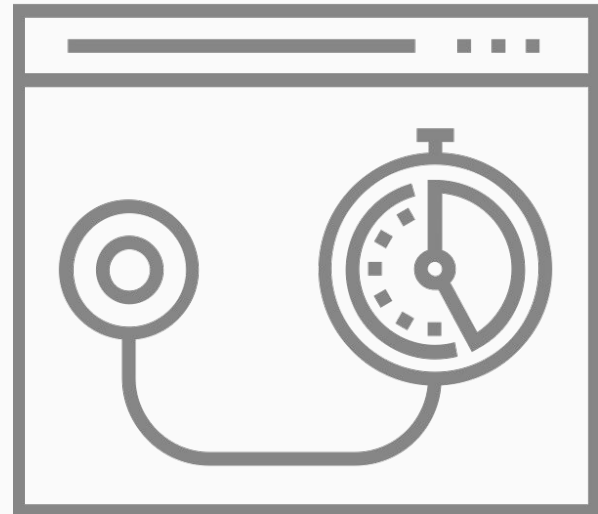
Why test?



It is fundamental for real world solutions!

Testing that...

- a procedure will return the right thing
- its performance is up to scratch
- the whole system works
- The code integrates with another system



DocTests

- Easy to write as you go
- Almost no setup
- Searches for things that look like Python sessions >>>

```
import doctest

def add(x, y):
    """
    Test
    Add two numbers together
    Expected
    >>> add(1, 3)
    4
    >>> add(5, 2)
    7
    """
    return x + y

doctest.testmod(verbose=True)
```

Unit Testing

- More thorough testing structure
- Allows for things to open on test start and end
- What I use to mark assignments!

```
import unittest

class TestStringMethods(unittest.TestCase):

    def test_upper(self):
        self.assertEqual('foo'.upper(), 'FOO')

    def test_isupper(self):
        self.assertTrue('FOO'.isupper())
        self.assertFalse('Foo'.isupper())

unittest.main()
```

Practical & Assignment 2



References

- <https://docs.python.org/3/library/doctest.html>
- <https://docs.python.org/3/library/unittest.html>
- <https://thenounproject.com/>