Python
Unit test

```
#!/usr/local/bin/python3
```

Run on command line

```
import unittest
import template
```

unit tests for file

file

```
class TestStuff(unittest.TestCase):
```

Class name unimportant. unittest.TestCase Vita

```
    def setUp(self):
        self.input = 'foo'
        self.t = template.Template(self.input)
```

To reuse data in tests

file class

```
    def test_calc(self):
        d = self.t.calc(False)
        self.assertEqual(self.input, d)

    def test_calc_pretty(self):
        pretty = 'One arg is: ' + self.input
        d = self.t.calc(True)
        self.assertEqual(pretty, d)

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

run all tests

Python Program

```
#!/usr/local/bin/python3                    run on command line

import argparse

class Template:                             Class w/constructor

    def __init__(self, datum):
        self.datum = datum

    def calc(self, pprint):                 Class method
        if pprint:
            result = "One arg is: " + self.datum
        else:
            result = self.datum
        print(result)
        return result                       Alway return (for testing)
                                                       Help string
if __name__ == '__main__':
    args_parser = argparse.ArgumentParser(description='Repeats command line
args',epilog='Nothing else.')
    args_parser.add_argument('-p', '--pprint', action='store_true',  — true/false flag
default=False)
    args_stuff = args_parser.parse_known_args()
    args = args_stuff[0]       — options
    data = args_stuff[1]       — ARGV

    for datum in data:                     Loop through data
        t = Template(datum)                  make objects
        t.calc(args.pprint)                  run functions
```