Python vs Perl: Tests

def ...
test_x(self)
if '__name__' = '__main__'

```
#!/usr/bin/env python3.3

import unittest
import Size

class SizeTest(unittest.TestCase):

    def setUp(self):
        self.foo_size = Size.Size(1)

    def test_s(self):
        self.assertEqual(self.foo_size.s, 1)

    def test_simple_print(self):
        sp_string = "1"
        sp = self.foo_size.simple_print()
        self.assertEqual(sp_string, sp)

    def test_pretty_print(self):
        pp_string = "The size is: 1."
        pp = self.foo_size.pretty_print()
        self.assertEqual(pp_string, pp)

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

my
new obj()
obj->func()
ok()

```
#!/usr/bin/env perl -w

use strict;
use FindBin qw($Bin);
use lib "$Bin/../..";
use Test::More tests=>3;
use Visualphysics::Layout::Size;

my $foo = new Visualphysics::Layout::Size(1, 2);

ok($foo->{s} == 1, "test s");

my $sp_string = "1";
my $sp = $foo->simple_print();
ok($sp_string eq $sp, "test simple_print()");

my $pp_string = "The size is: 1.";
my $pp = $foo->pretty_print();
ok($pp_string eq $pp, "test pretty_print()");
```