* Unittest是Python标准库中提供的单元测试工具
* 任何单元测试用例的类都需要继承unittest.TestCase类
* 测试用例函数命名必须以test\_开头
* unittest提供了两个特殊的方法：setUP和tearDown，它们会在类的每个测试方法调用前后执行一次

python提供了nose包来执行单元测试，通过nosetests命令，加载所有以test\_开头的文件，然后执行其中以test\_开头的函数（在windows下添加代码 unittest.main()， 就可执行测试代码）

由于在使用import包含一个模块时，这个模块可能会有测试代码，如果执行，就会将模块中的测试代码也一并执行，这样查看当前文件的测试信息时就会带来麻烦，如何处理？

在windows平台下

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

unittest库也提供了一个runner，可以使用$ python -m unittest test\_module的命令来执行某个模块的单元测试

在使用nosetests时，经常会只想运行某个特定的测试，具体语法为：path.to.your.module:ClassOfYourTest.test\_method

测试代码存放位置storage/impl\_mongodb.py的测试应该存储在tests/storage/test\_impl\_mongodb.py中

测试文件名称test\_之后的部分需要能体现出所测试的内容

非测试函数不以test开头，可以作为公共函数。

测试环境管理: tox

测试用例的运行和管理: testrepository, subunit, coverage

测试用例的编写: mock, testtools, fixtures

import unittest

def \_add(a, b):

"加法"

return a + b

class TestMe(unittest.TestCase):

def setUp(self):

print 'setUp'

def tearDown(self):

print 'tearDown'

def test\_add1(self):

self.assertEqual(\_add(1, 2), 3)

def test\_add2(self):

self.assertEqual(\_add(1, 2), 3)

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()