

# 一、Pytest使用简介

## 前言

pytest 是 python 的第三方单元测试框架，比自带 unittest 更简洁和高效，支持315种以上的插件，同时兼容 unittest 框架。这就使得我们在 unittest 框架迁移到 pytest 框架的时候不需要重写代码。接下来我们在文中来分析下 pytest 有哪些简洁、高效的用法。

## 环境搭建

首先使用 pip 安装 pytest

```
pip install pytest
```

```
pip install pytest-html 原生态报告模板
```

查看 pytest 是否安装成功

```
pip show pytest
```

1.创建 test\_sample.py 文件，代码如下：



```
1  #!/usr/bin/env python
2
3  # coding=utf-8
4
5  import pytest
6
7  def test_001():
8      assert 1 == 5
9
10 if __name__ == "__main__":
11     pytest.main()
```

执行结果：

```

1 test_sample.py F
  [100%]
2
3 ===== FAILURES =====
4 _____ test_answer _____
5
6     def test_001():
7
8     E         assert 1 == 5
9
10 test_sample.py:19: AssertionError
11 ===== 1 failed in 0.41s =====

```

从上面的例子可以看出，pytest 中断言的用法直接使用 `assert`，和 `unittest` 中断言 `self.assert` 用法有所区别。

**总结一下：使用 pytest 执行测试需要遵行的规则：**

- .py 测试文件必须以 `_test` 开头（或者以 `_test` 结尾）
- 测试类必须以 `Test` 开头，并且不能有 `init` 方法
- 测试方法必须以 `test_` 开头
- 断言必须使用 `assert`

**运行：pytest 用例路径 --html=./report/result.html 注意：--html= 没有空格。**

**还可以用 `main()` 方法来运行：**`**pytest.main(['当前用例路径', '--html=测试报告/XX.html '])**`

**在pytest中有四种setup和teardown，**

- 1、`setup_module`和`teardown_module`在整个测试用例所在的文件中所有的方法运行前和运行后运行，只会运行一次；
- 2、`setup_class`和`teardown_class`则在整个文件中的一个class中所有用例的前后运行，
- 3、`setup_method`和`teardown_method`在class内的每个方法运行前后运行，
- 4、`setup_function`、`teardown_function`则是在非class下属的每个测试方法的前后运行；

```

1 @pytest.mark.parametrize('inData, repsData', 列表)
2 if __name__ == '__main__':
3     pytest.main(['test_login.py', '-s', '--html=./res.html'])

```

**课上演示代码**

```

1 import pytest # 导入模块
2 def test_001(): # 函数
3     # 断言
4     print('-----test_001----- 开始')
5     assert 1+2 == 3
6     print('-----test_001----- 结束')
7
8 def test_002(): # 函数
9     # 断言
10    assert 1+2 == 0

```

```
11
12 #封装测试类
13 class Test_login:
14     def test_003(self):
15         # 断言
16         assert 1 + 2 == 4
17 #数据驱动
18 @pytest.mark.parametrize('inData,a',[(10,20),(30,40)])
19 def test_001(inData,a):#函数
20     #断言
21     print('-----test_001----- 开始')
22     assert inData+a == 3
23     print('-----test_001----- 结束')
24
25
26
27 if __name__ == '__main__':
28     pytest.main(['test_func.py', '-s', '--html=./report/xt.html'])#-s 输出信
29     息
```