

```

1  #pytest
2  #接口测试工具postman,jmeter,soupu,fiddler,bad boy,
3  #性能测试工具loadrunner,jmeter
4  #代码类, 现成框架unittest,pytest,nose,rf
5  #测试说白了, 就是比较预期结果和实际结果是否相同
6  #pytest框架使用的一些规定
7  #文件必须以test_*.py命名,或者*_test.py
8  #类必须以Test开头,且不能有__init__方法
9  #方法必须以test_开头,断言必须使用assert
10 #pytest的安装pip install pytest
11 #setup方法,在每个方法执行之前执行; teardown在每个方法全部执行完毕之后执行
12 import pytest
13 import os
14 class Test1:
15     # def teardown_method(self): #①
16     #     print('所有程序全部执行完毕')
17     # def teardown_class(self):
18     #     print('所有程序全部执行完毕')
19     # @pytest.mark.parametrize('a',[1,2,3]) #对变量a进行参数化
20     @pytest.mark.parametrize('a,b,c',[(1,2,3),(3,3,7),(2*2,2**2,8)]) #对多个
    变量进行参数化
21     def test_abc(self,a,b,c):
22         print('-----开始检测abc-----') #②
23         assert a+b==c
24     # def test_abc2(self):
25     #     print('-----开始检测abc2-----') #③
26     #     assert 2**25 == 67108864
27     # def setup_method(self):
28     #     print('前置条件开始运行') #④
29     # def setup_class(self):
30     #     print('前置条件开始运行') #④
31 if __name__ == '__main__':
32     pytest.main(['test_pytest0824.py','-s']) #-s表示显示print()的内容
33
34 #allure报告
35 import pytest
36 import os
37 import allure
38 #定义函数
39 @allure.feature('模块1') #allure.feature一级标签,allure.story二级标签
40 class TestLogin:
41     def setup_class(self):
42         print('-----前置条件-----')
43     @allure.story('页面1')
44     @allure.title('页面1标题')
45     @pytest.mark.parametrize('a',[1,2,3]) #参数化
46     def test_01(self,a):
47         print('-----01-----')
48         assert 1+2==a
49     @allure.title('页面1内容') #给allure报告加标题
50     def test_02(self):
51         print('-----02-----')
52         assert 1+2==6
53 @allure.feature('模块2')
54 class TestShopping:

```

```

55     @allure.story('模块2的页面1')
56     @allure.title('模块2的页面1具体内容')
57     @pytest.mark.parametrize('a,b', [(1,2),(3,4),(9,10)])
58     def test_02(self,a,b):
59         print('-----02-----')
60         assert a + 1 == b
61 if __name__ == '__main__':
62     #-s 输出打印信息
63     pytest.main(['test_fun1.py', '-s', '--alluredir', '../report']) #生成allure
报告需要的数据(任何报告都需要数据的支持)
64     os.system('allure generate ../report -o ../report/report --clean')
65
66 #setup_class,teardown_class ④23①
67 #setup_method,teardown_method ④2①④3①
68 #setup与teardown是可选方法，可以不使用，也可以只使用一个，也可以都使用
69
70 #数据驱动（参数化）
71 # @pytest.mark.parametrize('a,b,c',[(1,2,3),(3,3,7),(2*2,2**2,8)]) #对多个变
量进行参数化
72
73 #allure的环境搭建：1.下载allure.zip 2.解压并放到任意目录 3.在环境变量path中加入
allure的目录，加到\bin这一层
74 #4.安装pip install allure-pytest 安装allure支持pytest的定制化报告
75 #如果安装有问题，cmd里执行pip install -i
https://pypi.tuna.tsinghua.edu.cn/simple --trusted-host
pypi.tuna.tsinghua.edu.cn allure-pytest
76
77 #allure端口冲突的解决办法
78 #1.打开cmd窗口，输入netstat -ano|findstr 端口号(比如63342)，找到最右侧的数字{n}
79 #2.杀掉对应进程 taskkill /f /t /im {n}

```