

```

1  #test_pytest201019
2  #pytest框架的命名规范
3  #①文件应当以test_*.py命名,或者*_test.py
4  #②类以Test开头,且类当中不能有__init__方法
5  #③方法必须以test_开头
6  #④断言必须使用assert
7
8  # import pytest
9  # class Test1:
10 #     def teardown_method(self): #①
11 #         print('最后执行的方法')
12 #     def test_abc(self): #②
13 #         print('开始执行方法1')
14 #         assert 1==1
15 #     def test_bbc(self): #③
16 #         print('开始执行方法2')
17 #         assert 1 == 2
18 #     def setup_method(self): #④
19 #         print('最开始执行的方法')
20
21 #setup_method,在所有方法之前执行,而且在每个方法执行之前,都会执行一次
22 #teardown_method,在所有方法之后执行,而且在每个方法执行之后,都会执行一次
23 #④②①④③①
24 #setup_class,teardown_class
25 #④②③①
26
27 import pytest
28 import os
29 import allure
30 @allure.feature('模块1') #一级标签
31 class Test1:
32     #数据驱动(参数化)
33     @pytest.mark.parametrize('number1,number2,number3',
34 [(2**25,33554432,67108864),(3*3,27,36),(2**0.5,1.414,9)]) #声明下方开始进行参
35     #数化
36     def test_abc(self,number1,number2,number3):
37         print('开始执行方法1')
38         assert number1+number2==number3
39 @allure.feature('模块2') #一级标签
40 class Test2:
41     #数据驱动(参数化)
42     @pytest.mark.parametrize('number1,number2,number3',
43 [(2**25,33554432,67108864),(3*3,27,36),(2**0.5,1.414,9)]) #声明下方开始进行参
44     #数化
45     @allure.story('页面2_1')
46     @allure.title('标题2_1')
47     def test_abc(self,number1,number2,number3):
48         print('开始执行方法1')
49         assert number1+number2==number3
50
51     @allure.story('页面2_2')
52     @allure.title('标题2_2')
53     def test_bbc(self):
54         assert 3==6
55
56     @allure.story('页面2_9')

```

```
52     def test_cbc(self):
53         assert 4==5
54 if __name__ == '__main__':
55     pytest.main(['test_pytest201019.py', '-s', '--alluredir', '../report'])
    #-s表示执行各个方法中的print语句
56     os.system('allure generate ../report -o ../report/report --clean')
```

松勤SONGQIN