

# 三日精通：自动化必备之Pytest测试框架训练营

---

## 第二天：pytest测试框架核心技巧

---

主讲：北凡老师

### 上节回顾

---

## 8. 夹具fixture

---

夹具：在用例**执行之前**、**执行之后**，自动运行代码

场景：

- 之前：加密参数 / 之后：解密结果
- 之前：启动浏览器 / 之后：关闭浏览器
- 之前：注册、登录账号 / 之后：删除账号

### 1. 创建fixture

```
@pytest.fixture
def f():

    # 前置操作
    yield
    # 后置操作
```

1. 创建函数
2. 添加装饰器
3. 添加yield关键字

## 2. 使用fixture

1. 在用例的参数列表中，加入 fixture名字即可
2. 给用例加上 `usefixtures` 标记

```
def test_1(f):  
    pass  
  
@pytest.mark.usefixtures("f")  
def test_2():  
    pass
```

## 3. 高级用法

1. 自动使用
2. 依赖使用
  1. linux: 使用linux进行编译
  2. git: 使用git进行版本控制
  3. fixture: 使用fixture进行前后置自动操作
3. 返回内容: 接口自动化封装: 接口关联
4. 范围共享:
  - 默认范围: function
  - 全局范围: session
    - 使用 `conftest.py`

命名空间 -> 第三空间

## 9. 插件管理

pytest插件生态是pytest特别的优势之处。

插件分成两类:

- 不需要安装: 内置插件

- 需要安装：第三方插件

插件的启用管理：

- 启用：`-p abc`
- 禁用：`-p no:abc`

插件使用方式：

1. 参数
2. 配置文件
3. fixture
4. mark

## 10. 常用第三方插件

---

pytest有1400+ 插件： [https://docs.pytest.org/en/stable/reference/plugin\\_list.html](https://docs.pytest.org/en/stable/reference/plugin_list.html)

### 1. pytest-html

用途：生成HTML测试报告

安装：

```
pip install pytest-html
```

使用：

```
--html=report.html --self-contained-html
```

# report.html

Report generated on 14-Nov-2024 at 21:08:13 by [pytest-html](#) v4.1.1

## Environment


Python	3.12.0
Platform	Windows-11-10.0.22631-SP0
Packages	<ul style="list-style-type: none"><li>• pytest: 8.3.3</li><li>• pluggy: 1.5.0</li></ul>
Plugins	<ul style="list-style-type: none"><li>• html: 4.1.1</li><li>• metadata: 3.1.1</li><li>• order: 1.3.0</li><li>• rerunfailures: 14.0</li><li>• result-log: 1.2.2</li><li>• xdist: 3.6.1</li></ul>

## Summary

3 tests took 5 ms.

(Un)check the boxes to filter the results.

☐ 0 Failed, ☒ 3 Passed, ☐ 0 Skipped, ☐ 0 Expected failures, ☐ 0 Unexpected passes, ☒ 0

Result 	Test
Passed	tests/test_baili.py::test_3
Passed	tests/test_beifan.py::test_1
Passed	tests/test_beifan.py::test_2

## 2. pytest-xdist

用途：分布式执行

安装：

```
pip install pytest-xdist
```

使用：

```
-n N
```

- 只有在任务本身耗时较长，超出调用成本很多的时候，才有意义
- 分布式执行，有并发问题：资源竞争、乱序

### 3. pytest-rerunfailures

用途：用例失败之后，重新执行

安装：

```
pip install pytest-rerunfailures
```

使用：

```
--reruns 5 --reruns-delay 1
```

### 4. pytest-result-log

用途：把用例的执行结果记录到日志文件中

安装：

```
pip install pytest-result-log
```

使用：

```
log_file = ./logs/pytest.log
log_file_level = info
log_file_format = %(levelname)-8s %(asctime)s [%(name)s:%(lineno)s] : %(message)s
log_file_date_format = %Y-%m-%d %H:%M:%S

; 记录用例执行结果
result_log_enable = 1
; 记录用例分割线
result_log_separator = 1
; 分割线等级
result_log_level_separator = warning
; 异常信息等级
result_log_level_verbose = info
```

## 11. 企业级测试报告

allure 是一个测试报告框架

安装

```
pip install allure-pytest
```

配置

```
--alluredir=temps --clean-alluredir
```

生成报告

```
allure generate -o report -c temps
```

allure支持对用例进行**分组和关联**（敏捷开发术语）

@allure.epic	史诗	项目
@allure.feature	主题	模块
@allure.story	故事	功能
@allure.title	标题	用例

使用相同装饰器的用例，自动并入一组

```
@allure.epic('码尚教育自动化测试')
@allure.feature('pytest框架训练营')
@allure.story('mark标记和筛选')
@allure.title("实现筛选的用例")
@pytest.mark.ut
def test_a():
    pass

@allure.epic('码尚教育自动化测试')
@allure.feature('pytest框架训练营')
@allure.story('fixture前置和后置')
@allure.title("使用fixture的用例")
@pytest.mark.ut
def test_b():
    pass
```

## 12. Web自动化测试实战

pytest仅进行用例管理，**不会控制浏览器**，需要借助新的工具：**selenium**

1. 只了解selenium

## 2. 搜索关于selenium的pytest插件

```
@pytest.fixture()
def selenium():
    d = webdriver.Chrome()
    yield d

    d.quit()
```

