* To be done
* **我们可以使用pytest\_addoption 自定义命令行参数，比如说我们需要在不同的环境中去运行我们的测试用例，那么我们就可以通过命令行去传递不同的环境标识，去得到我们想要的测试结果：**
* Pytest logging

Allure

<https://blog.csdn.net/qq_41780297/article/details/114214694?spm=1001.2101.3001.6650.3&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-3.pc_relevant_default&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-3.pc_relevant_default&utm_relevant_index=6>

<https://blog.csdn.net/NoamaNelson/article/details/105676063?spm=1001.2101.3001.6650.2&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-2.pc_relevant_default&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-2.pc_relevant_default&utm_relevant_index=5>

[https://blog.csdn.net/ouyicnm/article/details/117427862?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522165126717116782391891688%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request\_id=165126717116782391891688&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~first\_rank\_ecpm\_v1~rank\_v31\_ecpm-11-117427862.142^v9^pc\_search\_result\_control\_group,157^v4^control&utm\_term=Restful+API+setup+and+teardown&spm=1018.2226.3001.4187](https://blog.csdn.net/ouyicnm/article/details/117427862?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522165126717116782391891688%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=165126717116782391891688&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-11-117427862.142%5ev9%5epc_search_result_control_group,157%5ev4%5econtrol&utm_term=Restful+API+setup+and+teardown&spm=1018.2226.3001.4187)

**python自动化（一）基础能力：1.python文件读取**

<https://blog.csdn.net/ouyicnm/article/details/114677976?spm=1001.2101.3001.6650.17&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBLOGCOLUMN%7Edefault-17.pc_relevant_aa&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBLOGCOLUMN%7Edefault-17.pc_relevant_aa&utm_relevant_index=22>

<https://blog.csdn.net/Rainloving/article/details/46721959?spm=1001.2101.3001.6650.7&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-7.pc_relevant_antiscanv2&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-7.pc_relevant_antiscanv2&utm_relevant_index=10>

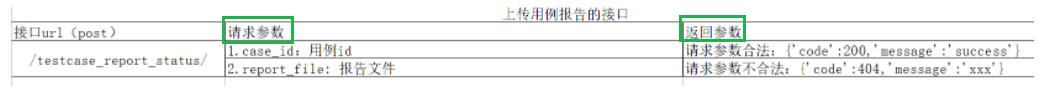
<https://blog.csdn.net/HG0724/article/details/122442019?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_title~default-4.pc_relevant_default&spm=1001.2101.3001.4242.3&utm_relevant_index=7>

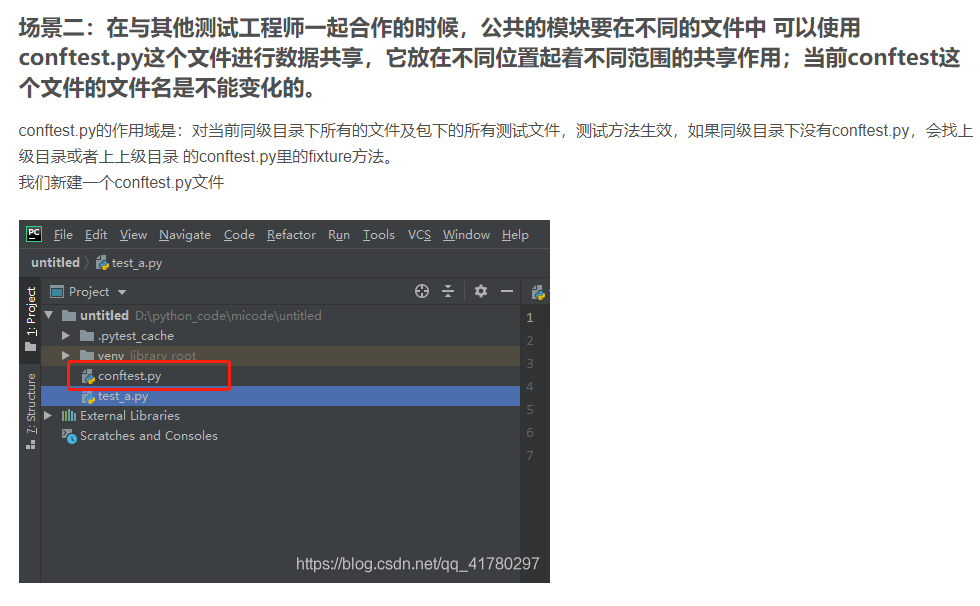
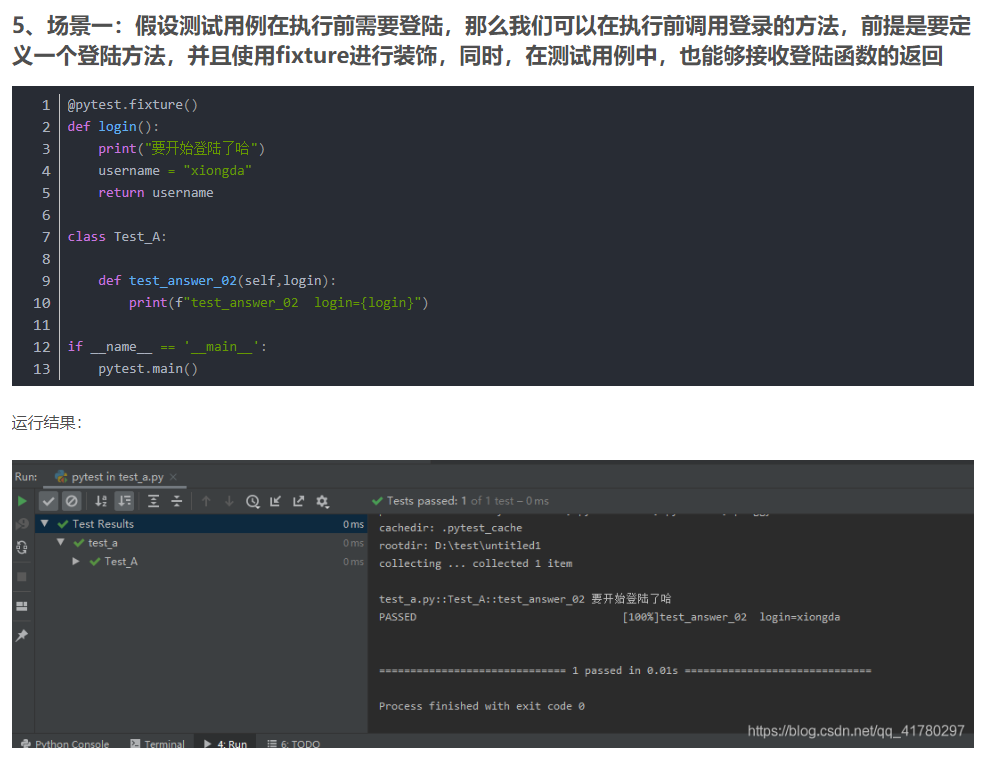
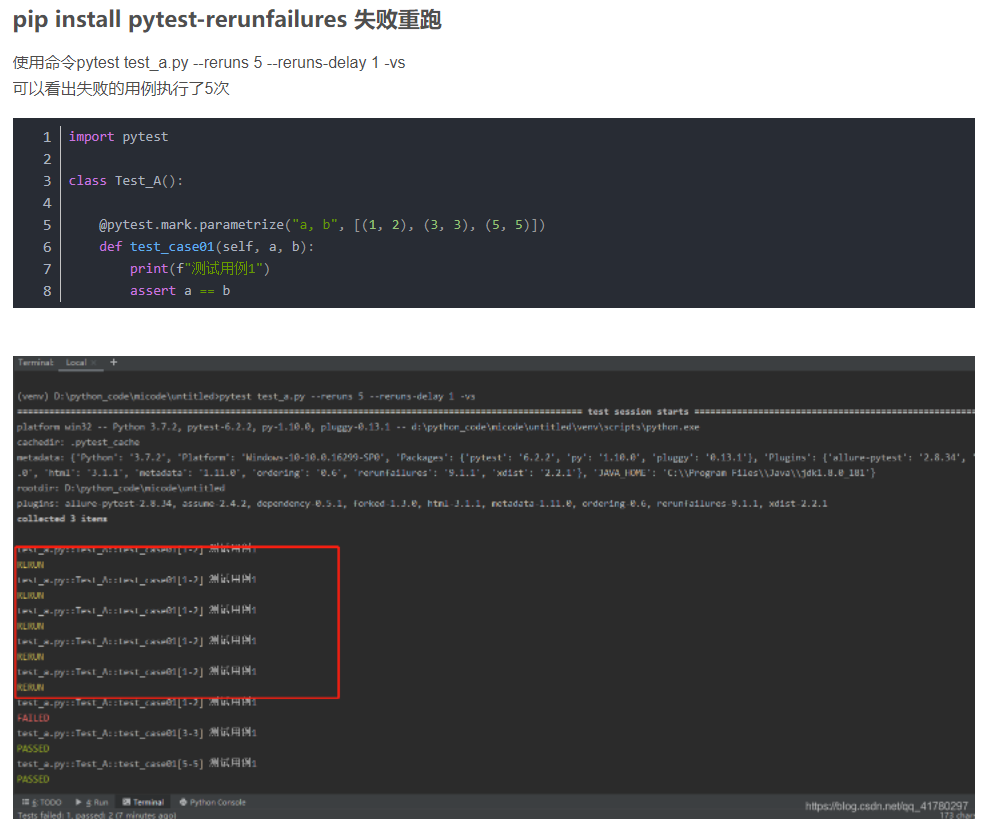
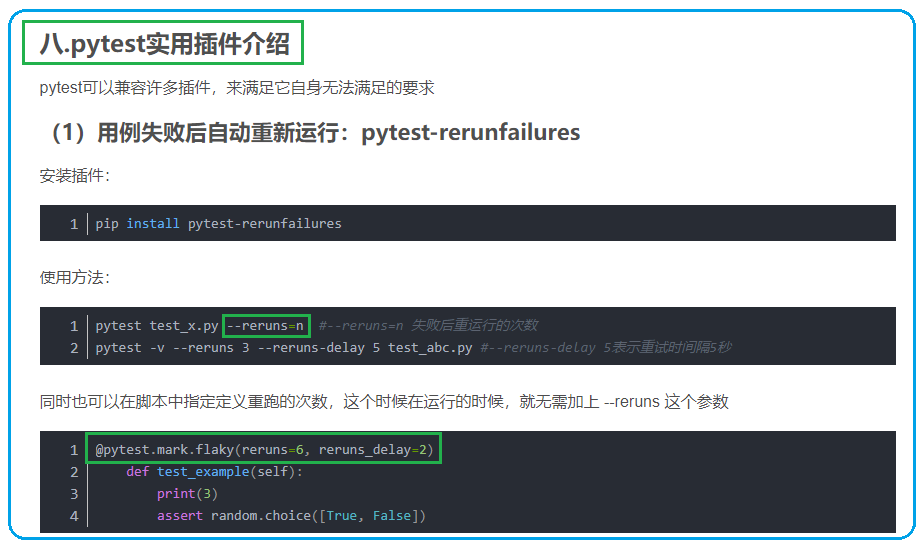
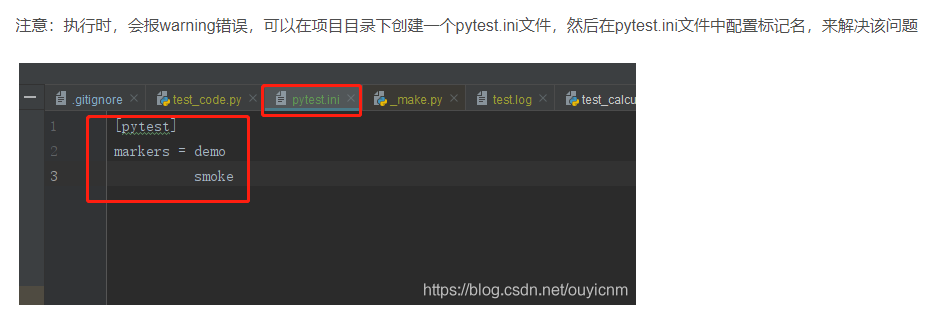
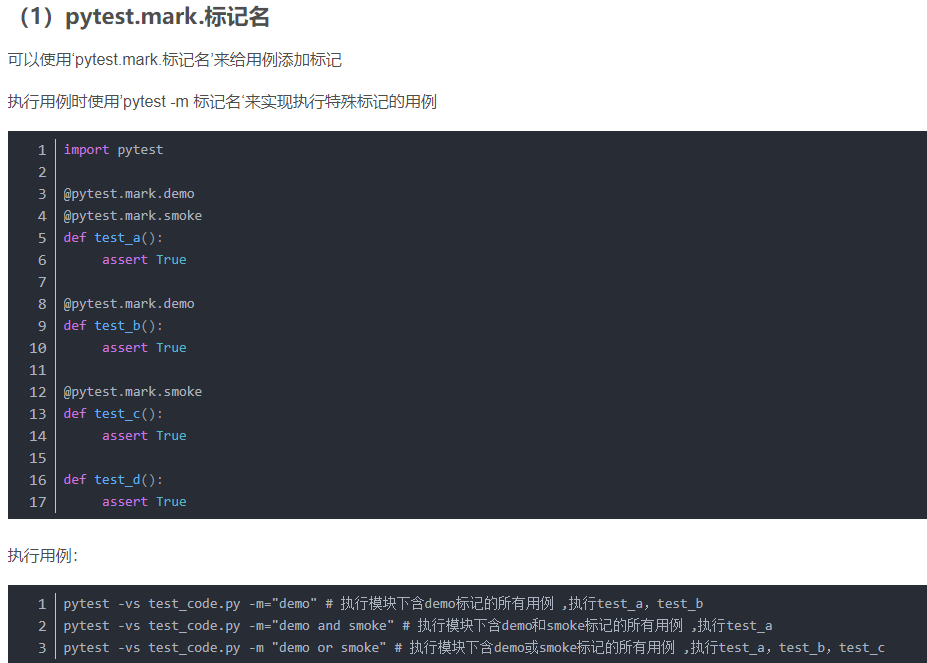
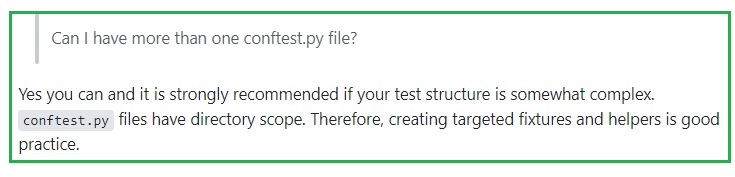
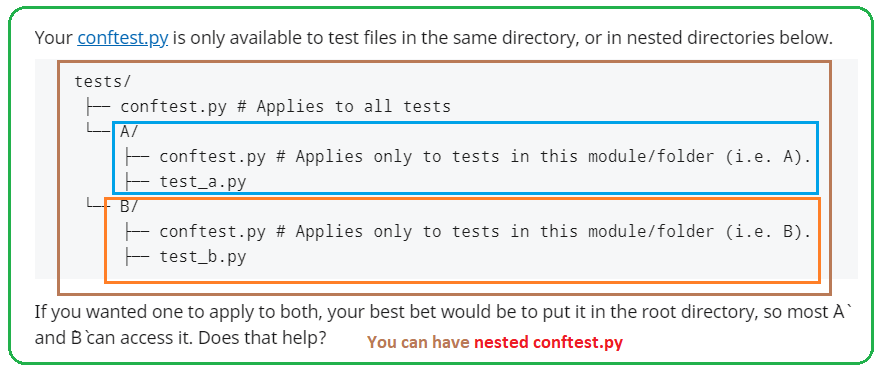
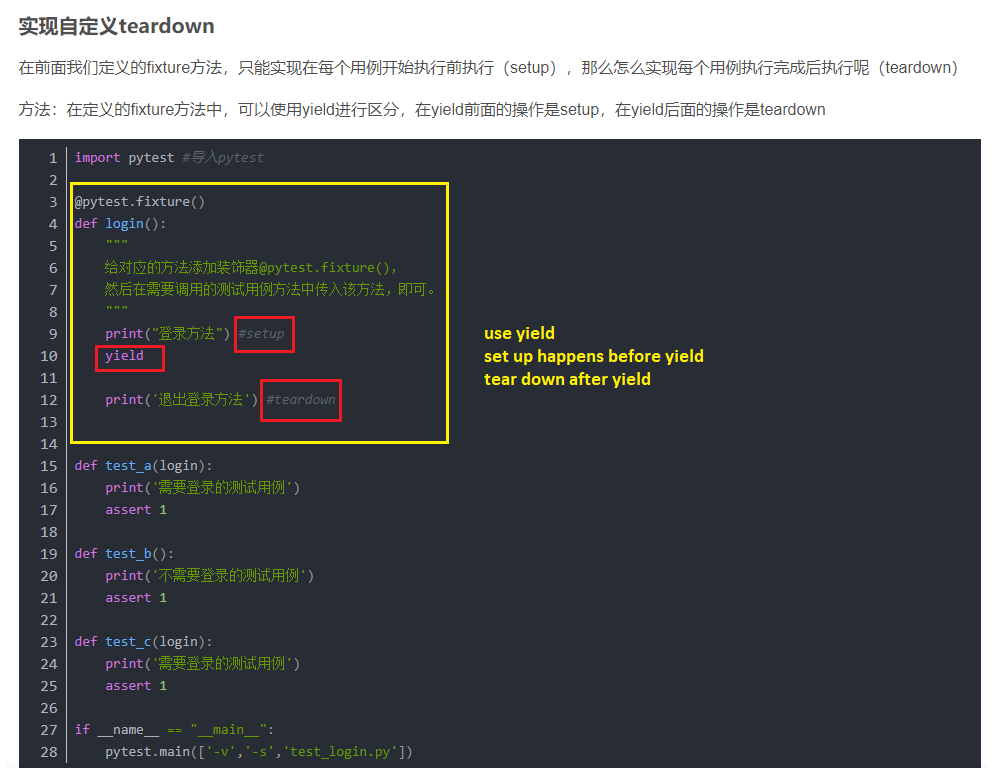
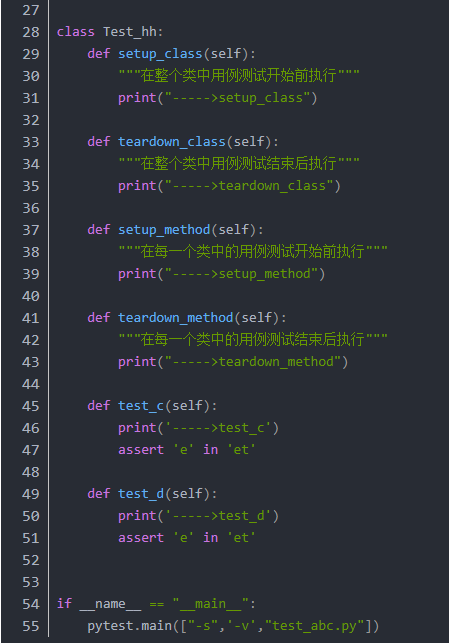
<https://blog.csdn.net/liangrui1988/article/details/49539137?_t=t&utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_title~default-5.pc_relevant_default&spm=1001.2101.3001.4242.4&utm_relevant_index=8>

<https://blog.csdn.net/u010624166/article/details/80132026?spm=1001.2101.3001.6650.16&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-16.pc_relevant_default&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-16.pc_relevant_default&utm_relevant_index=23>

# python自动化（三）web自动化：1.pytest框架讲解+集成allure

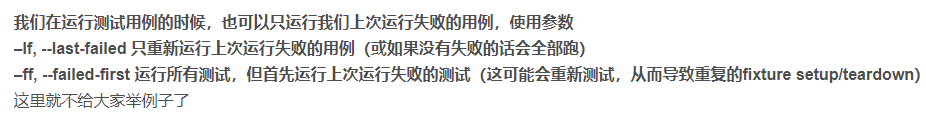
<https://blog.csdn.net/ouyicnm/article/details/116427973?spm=1001.2101.3001.6650.19&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBLOGCOLUMN%7Edefault-19.pc_relevant_aa&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBLOGCOLUMN%7Edefault-19.pc_relevant_aa&utm_relevant_index=24>







<https://blog.csdn.net/ouyicnm/article/details/117391676?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1.pc_relevant_aa&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1.pc_relevant_aa&utm_relevant_index=2>

  1.fixtures 语法

pytest.fixtures 的语法如下：

fixture(scope="function", params=None, autouse=False, ids=None, name=None)

从语法可以看到 fixture 的5个参数如下：

scope：用于控制 fixture 的作用范围

这个参数有以下4个级别：

function：在每一个 function 或者类方法中都会调用（默认）。

class：在每一个类中只调用一次。

module：每一个 .py 文件调用一次；该文件内可以有多个 function 和 class。

session：一个 session 调用一次。

**params**：一个可选的参数列表

params 以可选的参数列表形式存在。在测试函数中使用时，可通过 request.param 接收设置的返回值（即 params 列表里的值）。params 中有多少元素，在测试时，引用此 fixture 的函数就会调用几次。

autouse：是否自动执行设置的 fixtures

当 autouse 为 True 时，测试函数即使不调用 fixture 装饰器，定义的 fixture 函数也会被执行。

ids：指定每个字符串 id

当有多个 params 时，针对每一个 param，可以指定 id，这个 id 将变为测试用例名字的一部分。如果没有提供 id，则 id 将自动生成。

name：fixture 的名称

name 是 fixtures 的名称， 它默认是你装饰的那个 fixture 函数的名称。你可以通过 name 参数来更改这个 fixture 名称，更改后，如果这个 fixture 被调用，则使用你更改过的名称即可。  

* Pytest logging