为什么要用Pytest

1. 非常容易上手，入门简单，文档丰富
2. 支持参数化
3. 可以跳过某些测试用例
4. 支持重复执行(rerun)失败的 case
5. 支持运行由 unittest 编写的测试 case
6. 可生成 html 报告
7. 方便的和持续集成工具 jenkins 集成
8. 可支持执行部分用例
9. 具有很多第三方插件，并且可以自定义扩展

Pytest用例的设计原则

写用例时候，一定要按照下面的规则去写，否则不符合规则的测试用例是不会执行的

- 文件名以 test_*.py 文件和*_test.py
- 以 test_ 开头的函数
- 以 Test 开头的类，不能包含 __init__ 方法
- 以 test_ 开头的类里面的方法
- 所有的包 pakege 必须要有__init__.py 文件

Pytest 和 Unittest 的区别

比较点	pytest	unittest
用例编写	用例格式简单，兼容 unittest 用例	• 测试类必须继承 unittest.TestCase • 用例格式复杂，不兼容 pytest 格式
断言	使用 python 断言，也可以用断言插件	自带断言函数
用例前置后置	• 提供了共10种 teardown、setup 方法 • 还提供了 fixture 装饰器，更加强大	提供了共4种 teardown、setup 方法
参数化	自带参数化装饰器 @pytest.mark.parametrize	参数化需要用到 ddt 库
用例分类执行	• 可以通过@pytest.mark 标记测试用例 • pytest 执行命令加上 -m 参数运行指定标记的用例	通过加载不同的 testsuite 执行部分模块的用例
失败重运行	通过 pytest-rerunfailures 插件可以让失败的用例重跑	无
报告	自带的 pytest-html 报告，也可以结合 allure，更加美观	HTMLTestRunner 样式不咋滴，数据也不全

有几种 setup、teardown 方法?

10 种

有哪些级别的 setup、teardown 方法?

- 模块级别：setup_module、teardown_module
- 函数级别：setup_function、teardown_function，不在类中的方法
- 类级别：setup_class、teardown_class
- 类方法级别：setup_method、teardown_method
- 类方法细化级别：setup、teardown

fixture 是什么?

用来自定义测试用例的前置后置条件，不需要单独写 setup、teardown 方法

fixtrue 装饰器的默认作用域（scope）是什么级别?

function 方法级别

fixture 装饰器 scope 有哪几种值?

作用范围由小到大：function、class、module、package、session

不同 fixture 可以互相调用吗?

可以

测试用例如何调用 fixture? 有什么区别?

三种方式

- fixture 添加一个参数 autouse=True，用例会自动调用
- fixture 作为参数传入测试用例函数中
- 使用装饰器 @pytest.mark.userfixtures

一个测试用例调用了多个 fixture，调用顺序是怎么样?

范围越广的先调用，比如 session 级别的会先调用，然后再到 function 级别的被调用

如果要调用多个 fixture，应该怎么做? 装饰器嵌套的方式，可以实现吗?

- 一个测试用例可以调用多个 fixture，可以添加多个 @pytest.mark.userfixtures，也可以在函数参数列表那声明多个 fixture 名字
- 一个测试用例可以只调用一个 fixture，但这个 fixture 还会去调用其他更高层次的 fixture

pytest 实现 fixture 的原理是怎么样的?

- 当某个方法添加了 @pytest.fixture，会告知 pytest 这是一个 fixture
- 其他 fixture 或者测试用例可以调用它
- 它会在测试用例执行前后去运行
- （看了源码和官方文档，也没有说原理，有点尴尬）

fixture 如何实现 teardown 的效果?

- 在函数里面添加 yield 关键字
- 或添加一个 addfinalizer 函数

pytest 测试用例执行完后有哪几种状态?

- passed：测试通过
- failed：测试失败
- error：代码问题
- xfail：预期测试失败，但不会被当做是失败状态

如何跳过执行某些测试用例?

有两个装饰器

- @pytest.mark.skip

- @pytest.mark.skipif

conftest.py 文件是干嘛的?

- 存放 fixture 的文件
- 可以将一些全局通用的 fixture 放在里面，和测试用例解耦，比如：登录请求
- 一个项目可以有多个 conftest.py 文件
- 每个 conftest.py 只对当前 package 生效

pytest 怎么做参数化?

使用 @pytest.mark.parametrize

pytest.ini 文件是干嘛的?

- pytest 的主配置文件，可以改变 pytest 的默认行为
- pytest 每次运行前会读取该配置文件的内容，然后按指定的方式去运行
- 最常用的有：注册 mark 标签、默认命令行选项、修改 pytest 收集用例的目录、修改用例的规则

你用过哪些 pytest 插件?

- 失败重跑插件：pytest-rerunfailures
- HTML报告插件：pytest-html
- 重复执行用例：pytest-repeat
- 多重断言插件：pytest-assume
- 分布式测试插件：pytest-xdist
- 结合 allure 报告插件：allure-pytest

pytest-xdist 是线程并发还是进程并发?

进程并发

可以说下分布式测试的原理吗

- xdist 会产生一个或多个 workers
- workers 数量由参数 -n 决定，默认是 CPU 逻辑核数，每个 CPU 会启动一个进程跑测试用例集
- workers 都通过 master 来控制
- 每个 worker 复制执行完整的测试用例集，然后按照 master 的要求运行测试，而 master 机不执行测试任务

为什么会用 allure 作为测试报告?

- allure 报告展示了非常详细的测试数据，包括测试通过、测试失败、代码报错的数据整体统计、分类统计
- 管理层肯定会更加喜欢美观的测试报告
- 和 jenkins 可以完美的结合一起使用
- 可以自定义测试报告的一些内容

如何生成 allure 报告?

1. 下载 allure 的包
2. 解压，将 bin 目录添加到环境变量
3. 使用 pytest 命令执行时，添加 --alluredir 参数值即可

如何清空 allure 历史报告?

使用 pytest 命令执行时，添加 ––clean–alluredir 参数即可

如何动态生成测试用例的标题?

使用 @pytest.mark.parametrize 参数化，结合使用 @allure.title 装饰器，将标题值传入 title 装饰器中

有哪些常用的命令行参数

- -s：打印 print 的内容
- -q：简略打印
- -m：只运行指定 mark 标签的用例
- -k：运行指定关键字的测试用例
- -x：用例运行失败则立即停止执行
- --maxfail：用例运行时，允许做大失败次数，超过则停止运行
- --last-failed：只执行上次执行失败的测试用例

pytest 有哪些断言方式?

- 使用 python 的原生 assert 断言关键字
- 使用 pytest 的断言库 pytest-assume

",