

分析

python 的单元测试工具使用 unittest, unittest 的测试文件写在 .py 文件中, 在进行单元测试时调用写有单元测试的.py 文件即可, 经过调研不需要配置文件。

编写测试文件的基本思路

1. 先设置编码, utf-8 可支持中英文, 如上, 一般放在第一行;

2. 注释: 包括记录创建时间, 创建人, 项目名称;

3. 导入 unittest 模块 import unittest;

4. 定义测试类, 父类为 unittest.TestCase;

5. 定义 setUp() 方法用于测试用例执行前的初始化工作;

6. 定义测试用例, 以 "test_" 开头命名的方法

7. 定义 tearDown() 方法用于测试用例执行之后的善后工作。

8.1 执行测试用例方案一如下:

unittest.main() 方法会搜索该模块下所有以 test 开头的测试用例方法, 并自动执行它们, 执行顺序是命名顺序;

8.2 执行测试用例方案二如下:

8.2.1 先构造测试集

8.2.1.1 实例化测试套件

```
suite=unittest.TestSuite();
```

8.2.1.2 将测试用例加载到测试套件中

8.2.2 执行测试用例

8.2.2.1 实例化 TextTestRunner 类

```
runner=unittest.TextTestRunner();
```

8.2.2.2 使用 run() 方法运行测试套件 (即运行测试套件中的所有用例)

```
runner.run(suite);
```

8.3 执行测试用例方案三如下:

8.3.1 构造测试集 (简化了方案二中先要创建测试套件然后再依次加载测试用例)

执行顺序同方案一: 执行顺序是命名顺序: 先执行 test_case1, 再执行 test_case2

```
test_dir = './'
```

```
discover =unittest.defaultTestLoader.discover(test_dir,pattern='test_*.py');
```

8.3.2 执行测试用例

8.3.2.1 实例化 TextTestRunner 类

```
runner=unittest.TextTestRunner();
```

8.3.2.2 使用 run() 方法运行测试套件 (即运行测试套件中的所有用例)

```
runner.run(discover)。
```