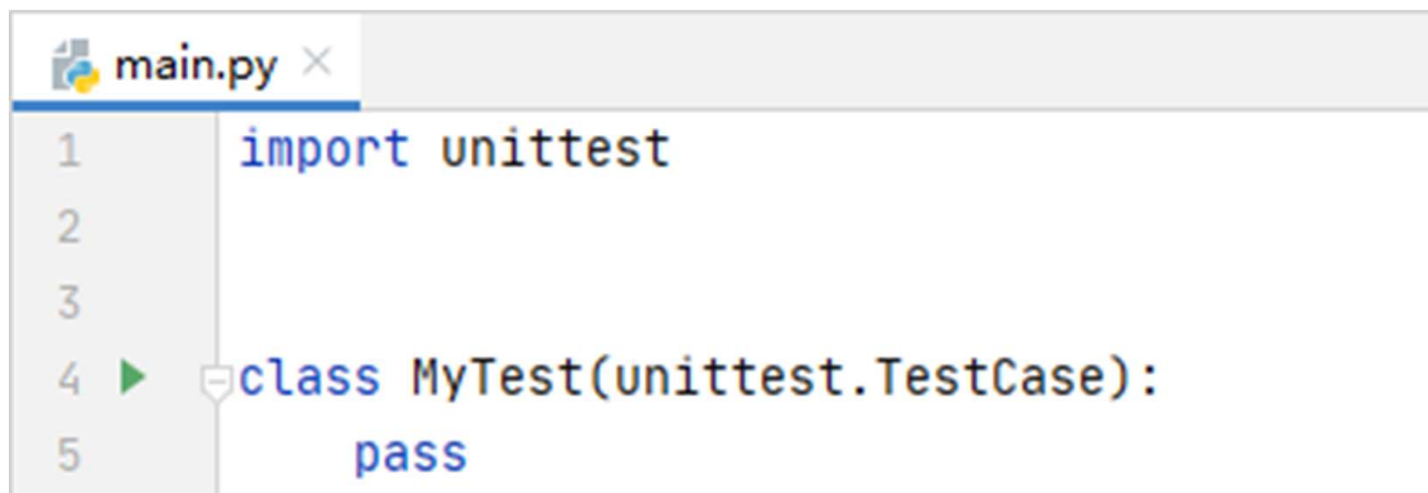


利用 unittest 进行单元测试

- unittest是Python单元测试框架，类似于Java的JUnit框架，主要用于Python代码单元测试，是一种优秀的测试框架。
- unittest包含以下四个概念：
 - test fixture： 对一个测试用例环境的搭建和销毁，是一个fixture，通过覆盖 TestCase的 setUp()和tearDown()方法来实现。
 - test case： 一个完整的测试单元。
 - test suite： 多个测试用例集合在一起，就是TestSuite，而且TestSuite也可以嵌套TestSuite。
 - test runner： 是来执行测试用例的，其中的run(test)会执行TestSuite/TestCase中的 run(result)方法

测试步骤:

- 1、编写一个python类，继承 unittest模块中的TestCase类，这就是一个测试类



The screenshot shows a code editor window titled 'main.py'. The code contains the following lines:

```
1 import unittest
2
3
4 class MyTest(unittest.TestCase):
5     pass
```

A green play button icon is visible next to line 4, indicating a run or execute action.

测试步骤:

- 2、在上面编写的测试类中定义测试方法（这个就是指的测试用例），每个方法的方法名要求以t打头，没有额外的参数。在该测试方法中调用被测试代码，校验测试结果，TestCase类中提供了很多标准的校验方法。

```
class MyTest(unittest.TestCase):  
    # 编写测试用例，记得要以t开头，例：test  
    def test(self):  
        # 调用内部的测试方法，如：assertEqual,assertNotEqual等  
        self.assertEqual(2 + 3, 5)
```

常见检验方法：（左侧为函数，右侧为测试通过的条件）

- `assertEqual(a, b)` `a == b`
- `assertNotEqual(a, b)` `a != b`
- `assertTrue(x)` `bool(x) is True`
- `assertFalse(x)` `bool(x) is False`
- `assertIs(a, b)` `a is b`
- `assertIsNot(a, b)` `a is not b`
- `assertIsNone(x)` `x is None`
- `assertIsNotNone(x)` `x is not None`
- `assertIn(a, b)` `a in b`
- `assertNotIn(a, b)` `a not in b`
- `assertIsInstance(a, b)` `isinstance(a, b)`
- `assertNotIsInstance(a, b)` `not isinstance(a, b)`

测试步骤:

- 3、执行 `unittest.main()`，该函数会负责运行测试，它会实例化所有 `TestCase` 的子类，并运行其中所有以 `test` 打头的方法。

```
# 运行测试
if __name__ == '__main__':
    unittest.main()
```

完整代码：

```
main.py x
1  import unittest
2
3
4  class MyTest(unittest.TestCase):
5      # 编写测试用例，记得要以t开头，例：test
6      def test(self):
7          # 调用内部的测试方法，如：assertEqual,assertNotEqual等
8          self.assertEqual(2 + 3, 5)
9
10
11     # 运行测试
12     if __name__ == '__main__':
13         unittest.main()
```