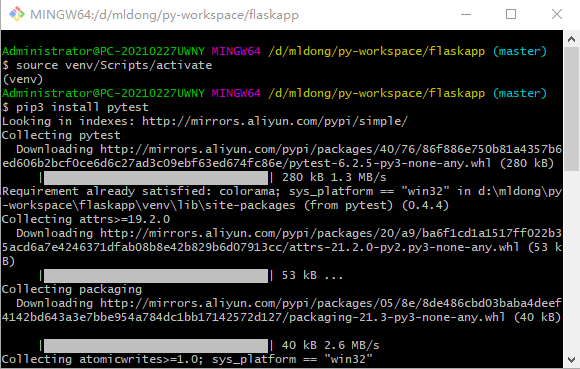
**11.1单元测试·pytest初探**

未经测试的小猫，不是一只好猫

前面的文章中，我们测试代码，基本上都是使用Postman来进行测试的，虽然使用上也不错，但是需要我们各层代码都写完，提供api接口后才能进行测试，这在我们代码开发过程中，并不是很方便，特别是面对一些复杂的业务的时候，就有些不方便了。最好的测试就是可以在各层代码中都能进行测试，也就是我们说的单元测试-指对软件中的最小可测试单元进行检查和验证。本文使用 [pytest](https://docs.pytest.org/" \t "https://pc.fenchuan8.com/" \l "/_blank) 包作为测试的基础框架，下面开始正式进入正题。

# 安装pytest

# 安装最新版本  
pip3 install pytest  
# 或指定安装版本  
pip3 install pytest==6.2.5



# 新建模块tests

mkdir tests

# 新增文件tests/\_\_init\_\_.py

空

# 新增文件tests/test\_demo1.py

# content of test\_demo1.py  
class TestDemo1:  
 def test\_one(self):  
 x = "this"  
 assert "h" in x  
  
 def test\_two(self):  
 x = "hello"  
 assert hasattr(x, "check")

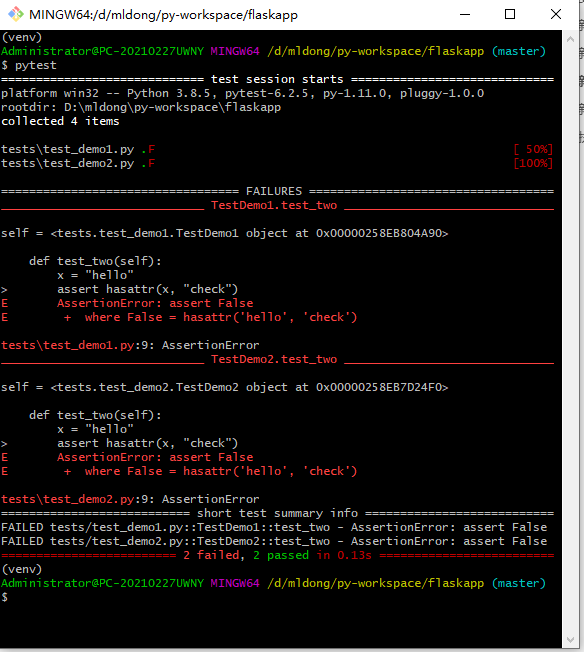
# 新增文件tests/test\_demo2.py

# content of test\_demo2.py  
class TestDemo2:  
 def test\_one(self):  
 x = "this"  
 assert "h" in x  
  
 def test\_two(self):  
 x = "hello"  
 assert hasattr(x, "check")

# 执行测试命令

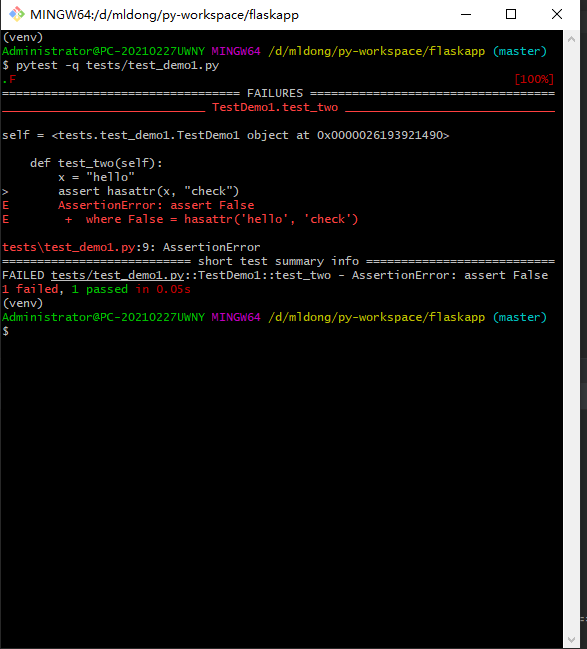
## 全部测试

pytest



## 指定测试某个文件

pytest -q tests/test\_demo1.py



# 重新修改代码调整-为正常

## 修改tests/test\_demo1.py

# content of test\_demo1.py  
class TestDemo1:  
 def test\_one(self):  
 x = "this"  
 assert "h" in x  
  
 def test\_two(self):  
 x = "hello"  
 assert not hasattr(x, "check")

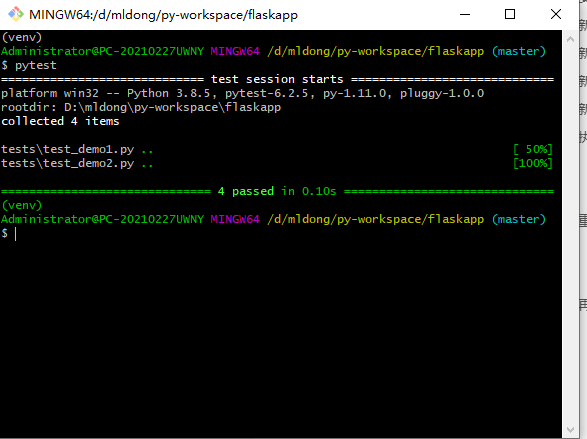
## 修改tests/test\_demo2.py

# content of test\_demo2.py  
class TestDemo2:  
 def test\_one(self):  
 x = "this"  
 assert "h" in x  
  
 def test\_two(self):  
 x = "hello"  
 assert not hasattr(x, "check")

# 再次执行测试命令

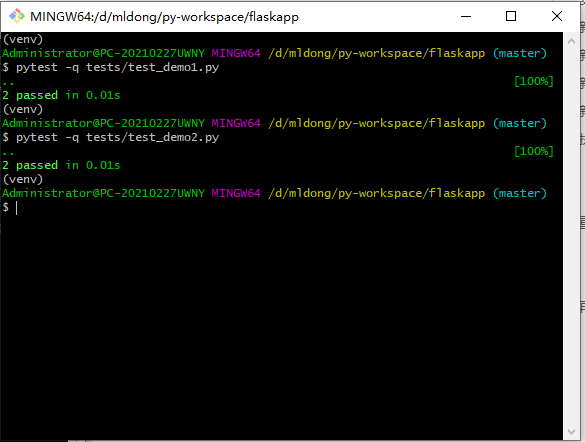
## 全部测试

pytest



## 指定测试某个文件

pytest -q tests/test\_demo1.py  
pytest -q tests/test\_demo2.py



# 一些说明

* 文件名称规范：以test\_开头
* 类名称：以Test开头
* 类测试方法：以test\_开头