

## 第4课 模块案例

# 课程概要

- ◆ 什么是虚拟环境
- ◆ 虚拟环境 `virtualenv`
- ◆ 虚拟环境 `pipenv`
- ◆ 实战-模块与包

# 课程目标

- ◆ 掌握虚拟环境的安装与使用
- ◆ 掌握python不同版本的虚拟环境的安装和使用
- ◆ 动手实现封装自己的模块和包

# 学习建议

◆ 多动手

◆ 多思考

◆ 多实践

# Python虚拟环境

# 什么是虚拟环境

- ◆ 思考：两个依赖不同Django版本的项目如何同时进行？

Django 1.11.x

Django 2.0



# 什么是虚拟环境

- ◆ 建立在宿主环境上的独立容器
- ◆ 具备和宿主环境相同的功能
- ◆ 快速创建和删除、方便管理

# 虚拟环境的好处

- ◆ 独立——相互隔离，互不影响
- ◆ 纯净——只有我一个项目用的包和依赖，好管理
- ◆ 方便——摒弃频繁安装/卸载包和依赖

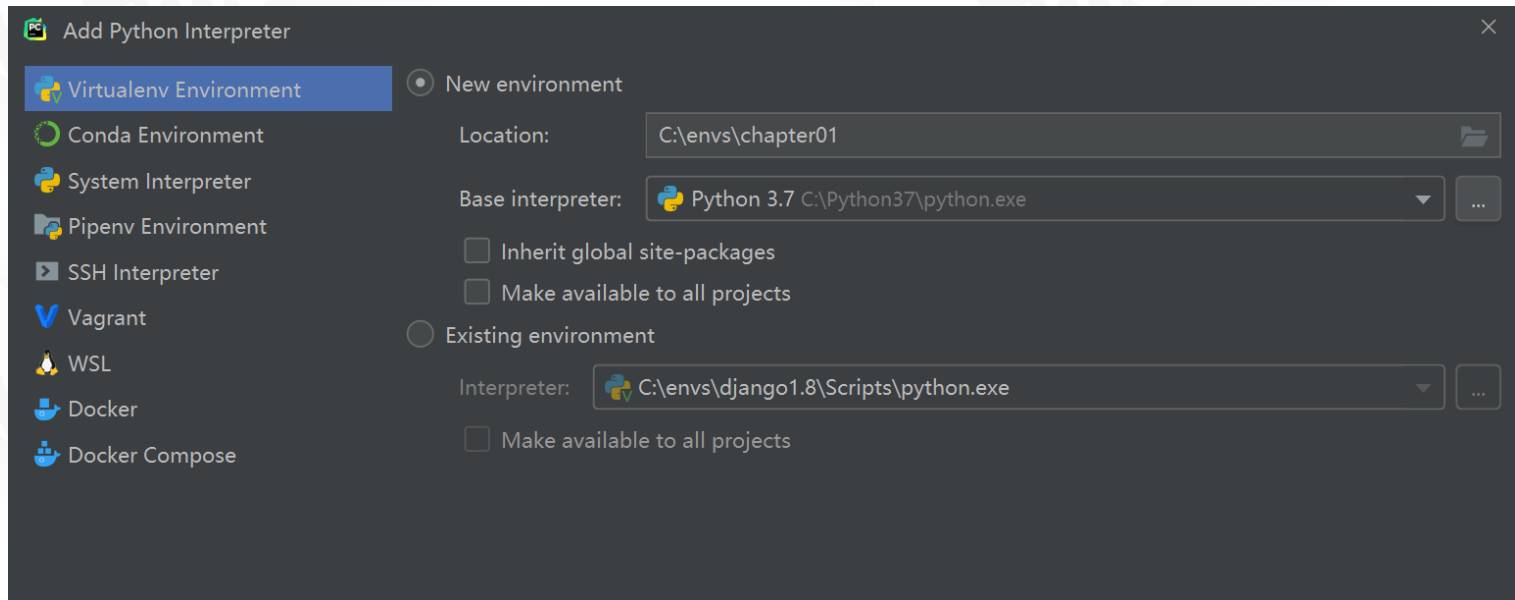


# 虚拟环境-virtualenv

- ◆ `pip install virtualenv`
- ◆ 进入/退出虚拟环境: `activate/deactivate`
- ◆ `pip install virtualenvwrapper-win`
- ◆ 虚拟环境自由切换: `workon vname`

# 虚拟环境-virtualenv

## ◆ 在pyCharm中使用虚拟环境



# 虚拟环境-pipenv

- ◆ pip install pipenv

- ◆ 创建及使用虚拟环境

```
mkdir py2.7  
cd py2.7  
pipenv --python 2.7  
pipenv install requests  
pipenv shell  
exit
```

# 实战—实现自己的工具包

# 实现自己的工具包

- ◆ 包含trans和work两个包
- ◆ 两个包都有同样的tools模块
- ◆ 包含constants常量模块
- ◆ 模块中会引用常量模块提供的常量值
- ◆ 在测试模块中可以分别调用tools相关方法

```
▼ chapter01
  ▼ utils
    ▼ trans
      ► __pycache__
        /* __init__.py
        /* tools.py
      ▼ work
        /* __init__.py
        /* tools.py
        /* __init__.py
        /* constants.py
        /* test_trans.py
```

# 重点知识

- ◆ 什么是虚拟环境
- ◆ 在什么场景下需要虚拟环境
- ◆ 如何安装和切换
- ◆ 模块与包的实战

## 难点知识

- ◆ 对虚拟环境的理解和使用
- ◆ 动手实践包与模块