: 你的第一本Python基础入门书 / 25 Python的影分身之术:虚拟环境

目录

第1章入门准备

01 开篇词: 你为什么要学 Python?

02 我会怎样带你学 Python?

03 让 Python 在你的电脑上安家落户

04 如何运行 Python 代码?

第2章通用语言特性

05 数据的名字和种类—变量和类型

06 一串数据怎么存—列表和字符串

07 不只有一条路—分支和循环

08 将代码放进盒子—函数

09 知错能改一错误处理、异常机制

10 定制一个模子—类

11 更大的代码盒子—模块和包

12 练习—密码生成器

第 3 章 Python 进阶语言特性

13 这么多的数据结构(一): 列表、元祖、字符串

14 这么多的数据结构(二):字典、

15 Python大法初体验:内置函数

16 深入理解下迭代器和生成器

17 生成器表达式和列表生成式

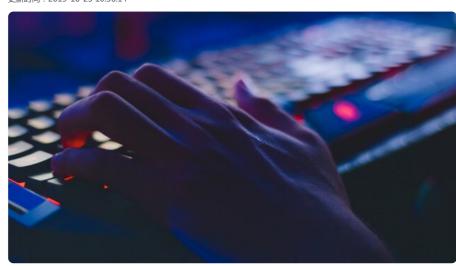
18 把盒子升级为豪宅:函数进阶

19 让你的模子更好用:类进阶

20 从小独栋升级为别墅区:函数式编

25 Python的影分身之术:虚拟环境

更新时间: 2019-10-23 10:36:14



今天应做的事没有做, 明天再早也是耽误了。

——裴斯泰洛齐

安装第三方包

在之前的文章中,我们多次使用了 Python 标准库,如 random 、 functools 。标准库直接被内置在 Python 中,无需安装。但标准库的功能也是有限的,有时并不足以满足我们的需求,这时可以考虑引入第三方包来解决。

Python 社区有大量的开发者和组织,贡献了数以万计的第三方包,它们包罗万象并且其中不乏高质量者,借助它们可以节省许多编码工作。Python 之所以开发效率高,一个原因就在于此。你可以在 Python Package Index 中查找所需要的包。

要安装第三方包可以使用命令行工具 pip ,它将从 Python Package Index 中检索并下载安装 所指定的包。

在命令行中使用如下命令:

pip3 install 包名

如安装 Python 下非常流行的 HTTP 客户端工具 requests:

pip3 install requests

→ pip3 install requests

: 你的第一本Python基础入门书 / 25 Python的影分身之术: 虚拟环境

目录

第1章入门准备

01 开篇词: 你为什么要学 Python?

02 我会怎样带你学 Python?

03 让 Python 在你的电脑上安家落户

04 如何运行 Python 代码?

第2章通用语言特性

05 数据的名字和种类—变量和类型

06 一串数据怎么存—列表和字符串

07 不只有一条路—分支和循环

08 将代码放进盒子—函数

09 知错能改一错误处理、异常机制

10 定制一个模子—类

11 更大的代码盒子—模块和包

12 练习—密码生成器

第3章 Python 进阶语言特性

13 这么多的数据结构(一):列表、 元祖、字符串

14 这么多的数据结构 (二):字典、 集合

15 Python大法初体验:内置函数

16 深入理解下迭代器和生成器

17 生成器表达式和列表生成式

18 把盒子升级为豪宅:函数进阶

19 让你的模子更好用:类进阶

20 从小独栋升级为别墅区:函数式编

https://files.pythonhosted.org/packages/51/bd/23c926cd341ea6b7dd0b2a00aba 99ae0f828be89d72b2190f27c11d4b7fb/requests-2.22.0-py2.py3-none-any.whl (57kB)

100% | 61kB 3.6kB/s

Collecting idna<2.9,>=2.5 (from requests)

Using cached

https://files.pythonhosted.org/packages/14/2c/cd551d81dbe15200be1cf41cd03869a46fe7226e7450af7a6545bfc474c9/idna-2.8-py2.py3-none-any.whl

Collecting urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 (from requests)

Downloading

 $https://files.pythonhosted.org/packages/e0/da/55f51ea951e1b7c63a579c09dd7\\ db825bb730ec1fe9c0180fc77bfb31448/urllib3-1.25.6-py2.py3-none-any.whl$ (125kB)

100% | 133kB 69kB/s

Collecting certifi>=2017.4.17 (from requests)

Downloading

https://files.pythonhosted.org/packages/18/b0/8146a4f8dd402f60744fa380bc7 3ca47303cccf8b9190fd16a827281eac2/certifi-2019.9.11-py2.py3-none-any.whl (154kB)

100% | 163kB 116kB/s

Collecting chardet<3.1.0,>=3.0.2 (from requests)

Using cached

https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1dd ec7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl

Installing collected packages: idna, urllib3, certifi, chardet, requests

Successfully installed certifi-2019.9.11 chardet-3.0.4 idna-2.8 requests-2.22.0 urllib3-1.25.6

安装成功后,就可以在代码中使用 import requests 来导入这个包了。

通过 pip 安装包时也可以指定包的版本:

pip3 install requests=2.22.0

也可以将一个已有的包升级到最新版本:

pip3 install --upgrade requests

pip 会将第三方包安装在 Python 目录下的 lib/python3.x/site-packages 目录中。如果项目代码中使用了某个第三方包,则 Python 会到这个目录中寻找。

可以使用 pip3 show 命令来查看第三方包的描述信息:

→ pip3 show requests

www.imooc.com/read/46/article/834

: 你的第一本Python基础入门书 / 25 Python的影分身之术:虚拟环境

目录

第1章入门准备

01 开篇词: 你为什么要学 Python?

02 我会怎样带你学 Python?

03 让 Python 在你的电脑上安家落户

04 如何运行 Python 代码?

第2章通用语言特性

05 数据的名字和种类—变量和类型

06 一串数据怎么存—列表和字符串

07 不只有一条路—分支和循环

08 将代码放进盒子—函数

09 知错能改一错误处理、异常机制

10 定制一个模子—类

11 更大的代码盒子—模块和包

12 练习—密码生成器

第3章 Python 进阶语言特性

13 这么多的数据结构(一):列表、元祖、字符串

14 这么多的数据结构(二):字典、

15 Python大法初体验:内置函数

16 深入理解下迭代器和生成器

17 生成器表达式和列表生成式

18 把盒子升级为豪宅:函数进阶

19 让你的模子更好用:类进阶

20 从小独栋升级为别墅区:函数式编

Summary: Python HTTP for Humans.

Home-page: http://python-requests.org

Author: Kenneth Reitz

Author-email: me@kennethreitz.org

License: Apache 2.0

Location: /Users/obsession/projects/venv/lib/python3.7/site-packages

Requires: certifi, chardet, idna, urllib3

Required-by:

虚拟环境

我们之前说过,使用 pip 安装的第三方包被集中放置在系统中 Python 目录下的 lib/python3. x/site-packages 目录中,这是个公共路径,其中的包可以被多个项目共享使用。

包被共享使用也会带来一个问题,如果不同的项目需要不同版本的包,这时就产生了版本冲突。比如:项目 1 中需要使用 requests=2.6.0 ,项目 2 中需要使用 requests=2.22.0 ,但 lib/pyt hon3.x/site-packages 目录中只能同时存在一个 requests 包的版本,这就不好办了!

同时,第三方包被集中放置也会带来一个问题,那就是如果系统中所安装的包数目非常大,对这 些包的管理也会是个问题。

这就引入了虚拟环境的概念了。

当我们拥有多个项目时,可以对这些项目分别创建单独的虚拟环境,每个虚拟环境就是一个专有的 Python 运行环境。各个项目运行在自己的虚拟环境中,被完全隔离开。当我们在一个虚拟环境中安装第三方包,这个包只能在当前虚拟环境中使用,其它虚拟环境中是无法看到它的,所以也就不会再有项目间版本冲突的问题。另外每个虚拟环境中只包含当前项目所使用到的包,不会包含大量无用的包。

创建虚拟环境

Python 3 中自带有虚拟环境模块,通过它即可创建虚拟环境,无需借助其它工具。

创建虚拟环境使用命令:

python3 -m venv 虚拟环境名称

如:

python3 -m venv venv

此时将在当前目录中创建出 venv 目录,这即是虚拟环境目录,其中包含 Python 运行环境。

激活虚拟环境

要使用虚拟环境首先需要激活它,激活后当前命令行所执行的 Python 代码都将运行于该虚拟环境中。

www.imooc.com/read/46/article/834

← 慕课专栏	: ■ 你的第一本Python基础入门书 / 25 Python的影分身之术:虚拟环境
目录	cd venv\Scripts activate.bat
第1章 入门准备	Linux 和 MacOS 中使用命令:
01 开篇词: 你为什么要学 Python ?	source venv/bin/activate
02 我会怎样带你学 Python?	上述命令执行过后,命令行的提示符前将显示(venv),如下:
03 让 Python 在你的电脑上安家落户	
04 如何运行 Python 代码?	(venv) →
第 2 章 通用语言特性	z表示当前已处于虚拟环境中。此后在该命令行中所执行的所有 Python 代码都运行于该虚拟环
05 数据的名字和种类—变量和类型	境中。
06 一串数据怎么存—列表和字符串	退出虚拟环境
07 不只有一条路—分支和循环	当我们不要使用虚拟环境时,可以退出当前虚拟环境。 Windows 操作系统中使用命令:
08 将代码放进盒子—函数	
09 知错能改一错误处理、异常机制	cd venv\Scripts deactivate.bat
10 定制一个模子—类	Linux 和 MacOS 中使用命令:
11 更大的代码盒子—模块和包	deactivate
12 练习一密码生成器	此后命令行的提示符将恢复回原本状态。
第 3 章 Python 进阶语言特性	 ← 24 让你的代码更灵活: 进程和线 程 26 更加 Python 的 Python 代码 风格
13 这么多的数据结构(一):列表、 元祖、字符串	
14 这么多的数据结构(二):字典、 集合	精选留言 0
15 Python大法初体验:内置函数	欢迎在这里发表留言,作者筛选后可公开显示
16 深入理解下迭代器和生成器	
17 生成器表达式和列表生成式	•
18 把盒子升级为豪宅:函数进阶	目前暂无任何讨论
19 让你的模子更好用:类进阶	

www.imooc.com/read/46/article/834

20 从小独栋升级为别墅区:函数式编

: 你的第一本Python基础入门书 / 25 Python的影分身之术:虚拟环境

目录

第1章入门准备

- 01 开篇词: 你为什么要学 Python?
- 02 我会怎样带你学 Python?
- 03 让 Python 在你的电脑上安家落户
- 04 如何运行 Python 代码?

第2章通用语言特性

- 05 数据的名字和种类—变量和类型
- 06 一串数据怎么存—列表和字符串
- 07 不只有一条路—分支和循环
- 08 将代码放进盒子—函数
- 09 知错能改—错误处理、异常机制
- 10 定制一个模子—类
- 11 更大的代码盒子—模块和包
- 12 练习—密码生成器

第 3 章 Python 进阶语言特性

- 13 这么多的数据结构(一):列表、 元祖、字符串
- 14 这么多的数据结构 (二):字典、
- 15 Python大法初体验:内置函数
- 16 深入理解下迭代器和生成器
- 17 生成器表达式和列表生成式
- 18 把盒子升级为豪宅:函数进阶
- 19 让你的模子更好用:类进阶
- 20 从小独栋升级为别墅区:函数式编

www.imooc.com/read/46/article/834