Python编程新思维及实战

嵩 天

Python第三方库安装方法

嵩天

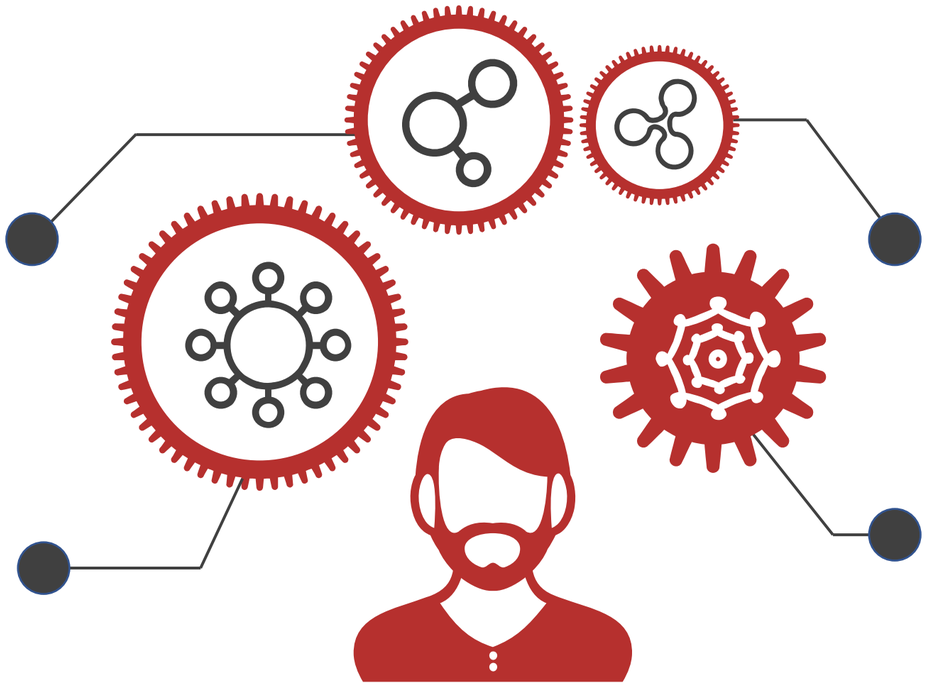
Python第三方库安装方法

单元开篇

单元开篇

(1) 第三方库的pip安装方法

(2) 第三方库的集成安装方法



(3) 第三方库的UCI安装方法

(4) 自动化pip安装脚本

Python第三方库安装方法

Python第三方库安装方法

pip安装方法

Python123

pip安装方法

pip是Python解释器自带的第三方库安装工具

- pip是命令行工具，适用于各种操作系统
- pip安装方法需要联网或给定安装文件
- 绝大部分Python库可使用pip安装，最主要的安装方法

pip安装方法

```
D:\>pip -h
```

Usage:

pip <command> [options]

Commands:

install	Install packages.
download	Download packages.
uninstall	Uninstall packages.
freeze	Output installed packages in requirements format.
list	List installed packages.
show	Show information about installed packages.
check	Verify installed packages have compatible dependencies.
search	Search PyPI for packages.
wheel	Build wheels from your requirements.
help	Show help for commands.

使用pip安装工具（命令行下执行）

Windows

Mac OS

Linux

pip安装方法

```
D:\>pip install <第三方库名或文件名>
```

- 安装指定的第三方库

```
pip install jieba   或   pip   install jieba-0.39.zip
```

pip安装方法

```
D:\>pip install -U <第三方库名>
```

- 使用-U标签更新已安装的指定第三方库

```
pip install -U jieba
```

pip安装方法

```
D:\>pip uninstall <第三方库名>
```

- 卸载指定的第三方库

```
pip uninstall jieba
```

pip安装方法

```
D:\>pip download <第三方库名>
```

- 下载但不安装指定的第三方库

```
pip download jieba
```

pip安装方法

```
D:\>pip show <第三方库名>
```

- 列出某个指定第三方库的详细信息

```
pip show jieba
```

pip安装方法

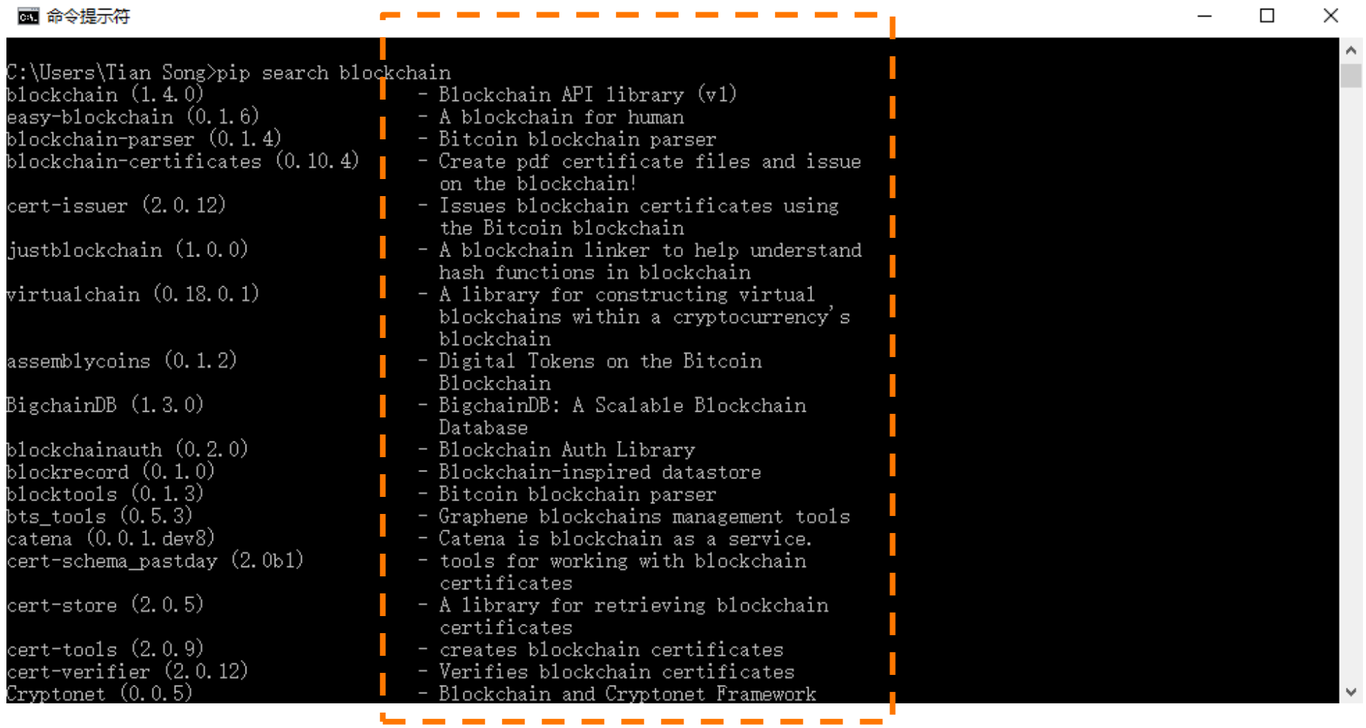
```
D:\>pip search <关键词>
```

- 根据关键词在名称和介绍中搜索第三方库

```
pip search blockchain
```

pip安装方法

```
D:\>pip search blockchain
```



pip安装方法

```
D:\>pip list
```

- 列出当前系统已经安装的第三方库

```
pip list
```

pip安装方法小结

pip是最主要的第三方库安装方法

二级命令：

-h、install、install -U、uninstall、

download、show、search、list

Python第三方库安装方法

集成安装方法

Python123

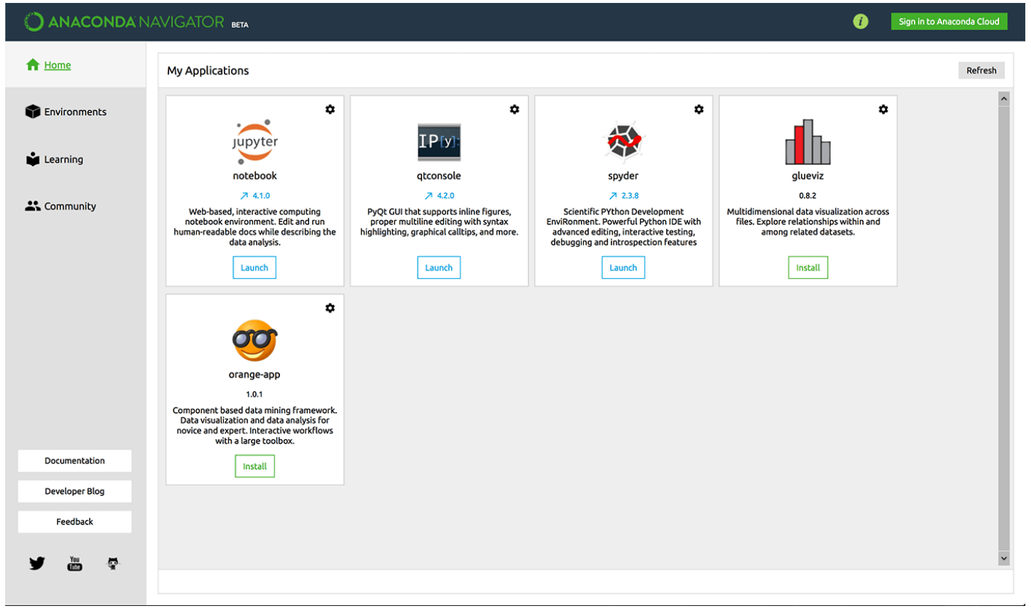
集成安装方法

集成安装：结合特定Python开发工具的批量安装

Anaconda

https://www.continuum.io

- 支持近800个第三方库
- 包含多个主流工具
- 适合数据计算领域开发



Python第三方库安装方法

UCI安装方法

Python123

UCI安装方法

pip不能解决100%的安装问题

- 某些第三方库在pip下载后，需要编译再安装，本地环境或没有编译器
- pip可能成功下载，但无法安装
- 对于这些特定库(小众)，是否有编译后的版本供下载安装呢?

UCI安装方法

http://www.lfd.uci.edu/~gohlke/pythonlibs/

Unofficial Windows Binaries for Python Extension Packages

by **Christoph Gohlke**, **Laboratory for Fluorescence Dynamics**, **University of California, Irvine**.

This page provides 32- and 64-bit Windows binaries of many scientific open-source extension packages for the official CPython distribution of the Python programming language.

The files are unofficial (meaning: informal, unrecognized, personal, unsupported, no warranty, no liability, provided "as is") and made available for testing and evaluation purposes.

Most binaries are built from source code found on PyPI or in the projects public revision control systems. Source code changes, if any, have been submitted to the project maintainers or are included in the packages.

Refer to the documentation of the individual packages for license restrictions and dependencies.

If downloads fail, reload this page, enable JavaScript, disable download managers, disable proxies, clear cache, use Firefox, reduce number and frequency of downloads. Please only download files manually as needed.

Use pip version 9 or newer to install the downloaded .whl files. This page is not a pip package index.

Many binaries depend on numpy-1.13+mkl and the Microsoft Visual C++ 2008 (x64, x86, and SP1 for CPython 2.7), Visual C++ 2010 (x64, x86, for CPython 3.4), or the Visual C++ 2017 (x64 or x86 for CPython 3.5, 3.6, and 3.7) redistributable packages.

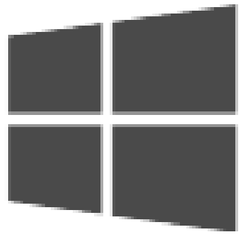
Install numpy+mkl before other packages that depend on it.

The binaries are compatible with the most recent official CPython distributions on Windows >=6.0. Chances are they do not work with custom Python distributions included with Blender, Maya, ArcGIS, OSGeo4W, ABAQUS, Cygwin, Pythonxy, Canopy, EPD, Anaconda, WinPython etc. Many binaries are not compatible with Windows XP or Wine.

The packages are ZIP or 7z files, which allows for manual or scripted installation or repackaging of the content.

The files are provided "as is" without warranty or support of any kind. The entire risk as to the quality and performance is with you.

The opinions or statements expressed on this page should not be taken as a position or endorsement of the Laboratory for Fluorescence Dynamics or the University of California.



Windows

UCI安装方法

UCI安装方法的三个步骤

- 步骤1：在UCI页面上搜索对应的第三方库
- 步骤2：下载对应Python解释器的编译后版本
- 步骤3：本地安装方法：pip install <下载文件名>

Python第三方库安装方法

pip安装脚本

Python123

实例需求

自动安装一批第三方库

- 具体需求：安装20个第三方库
- 自动运行：自动联网、自动安装，进行简单的错误判断

实例需求

拟安装的第三方库列表

库名	用途	安装指令
NumPy	N维数据表示和运算	pip install numpy
Matplotlib	二维数据可视化	pip install matplotlib
PIL	图像处理	pip install pillow
Scikit-Learn	机器学习和数据挖掘	pip install sklearn
Requests	HTTP协议访问及网络爬虫	pip install requests

实例需求

拟安装的第三方库列表

库名	用途	安装指令
Jieba	中文分词	pip install jieba
Beautiful Soup	HTML和XML解析器	pip install beautifulsoup4
Wheel	Python第三方库文件打包工具	pip install wheel
PyInstaller	打包Python源文件为可执行文件	pip install pyinstaller
Django	Python最流行的Web开发框架	pip install django

实例需求

拟安装的第三方库列表

库名	用途	安装指令
Flask	轻量级Web开发框架	pip install flask
WeRoBot	微信机器人开发框架	pip install werobot
SymPy	数学符号计算工具	pip install sympy
Pandas	高效数据分析和计算	pip install pandas
Networkx	复杂网络和图结构的建模和分析	pip install networkx

实例需求

拟安装的第三方库列表

库名	用途	安装指令
PyQt5	基于Qt的专业级GUI开发框架	pip install pyqt5
PyOpenGL	多平台OpenGL开发接口	pip install pyopengl
PyPDF2	PDF文件内容提取及处理	pip install pypdf2
docopt	Python命令行解析	pip install docopt
PyGame	简单小游戏开发框架	pip install pygame

代码详解

```
#BatchInstall.py
import os
libs = {"numpy","matplotlib","pillow","sklearn","requests",\
        "jieba","beautifulsoup4","wheel","networkx","sympy",\
        "pyinstaller","django","flask","werobot","pyqt5",\
        "pandas","pyopengl","pypdf2","docopt","pygame"}
try:
    for lib in libs:
        os.system("pip install " + lib)
    print("Successful")
except:
    print("Failed Somehow")
```

代码详解

```
#BatchInstall.py
import os
libs = {"numpy","matplotlib","pillow","sklearn","requests",\
        "jieba","beautifulsoup4","wheel","networkx","sympy",\
        "pyinstaller","django","flask","werobot","pyqt5",\
        "pandas","pyopengl","pypdf2","docopt","pygame"}
try:
    for lib in libs:
        os.system("pip install " + lib)
    print("Successful")
except:
    print("Failed Somehow")
```

利用os库函数启动pip命令进行安装

代码详解

```
#BatchInstall.py
import os
libs = {"numpy","matplotlib","pillow","sklearn","requests",\
        "jieba","beautifulsoup4","wheel","networkx","sympy",\
        "pyinstaller","django","flask","werobot","pyqt5",\
        "pandas","pyopengl","pypdf2","docopt","pygame"}
try:
    for lib in libs:
        os.system("pip install " + lib)
    print("Successful")
except:
    print("Failed Somehow")
```

待安装的库

代码详解

```
#BatchInstall.py
import os
libs = {"numpy","matplotlib","pillow","sklearn","requests",\
        "jieba","beautifulsoup4","wheel","networkx","sympy",\
        "pyinstaller","django","flask","werobot","pyqt5",\
        "pandas","pyopengl","pypdf2","docopt","pygame"}
try:
    for lib in libs:
        os.system("pip install " + lib)
    print("Successful")
except:
    print("Failed Somehow")
```

简单的异常处理

无法获得安装状态

代码详解

```
#BatchInstall.py
import os
libs = {"numpy","matplotlib","pillow","sklearn","requests",\
        "jieba","beautifulsoup4","wheel","networkx","sympy",\
        "pyinstaller","django","flask","werobot","pyqt5",\
        "pandas","pyopengl","pypdf2","docopt","pygame"}
try:
    for lib in libs:
        os.system("pip install " + lib)
    print("Successful")
except:
    print("Failed Somehow")
```

逐一安装第三方库

Python第三方库安装方法

单元小结

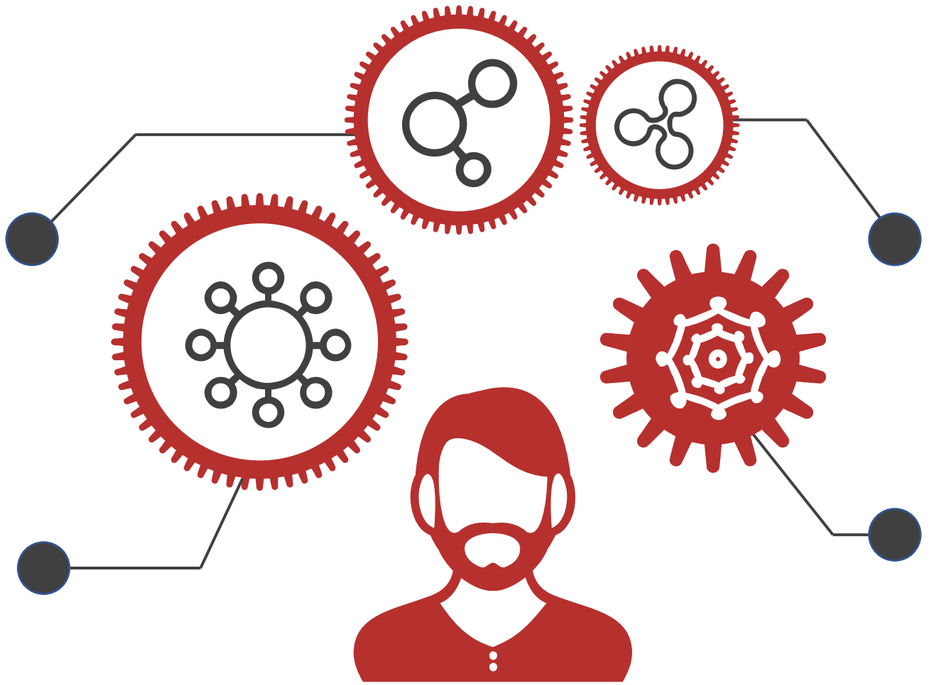
单元小结

(1) 第三方库的pip安装方法

pip的各种二级命令

(2) 第三方库的集成安装方法

anaconda等集成环境



(3) 第三方库的UCI安装方法

第三方库的UCI网站

(4) 自动化pip安装脚本

利用os.system()执行pip脚本

Python第三方库安装方法


Python 123
Thank you