

全国计算机等级考试二级教程

Python语言程序设计

(2018年版)



【第10章】

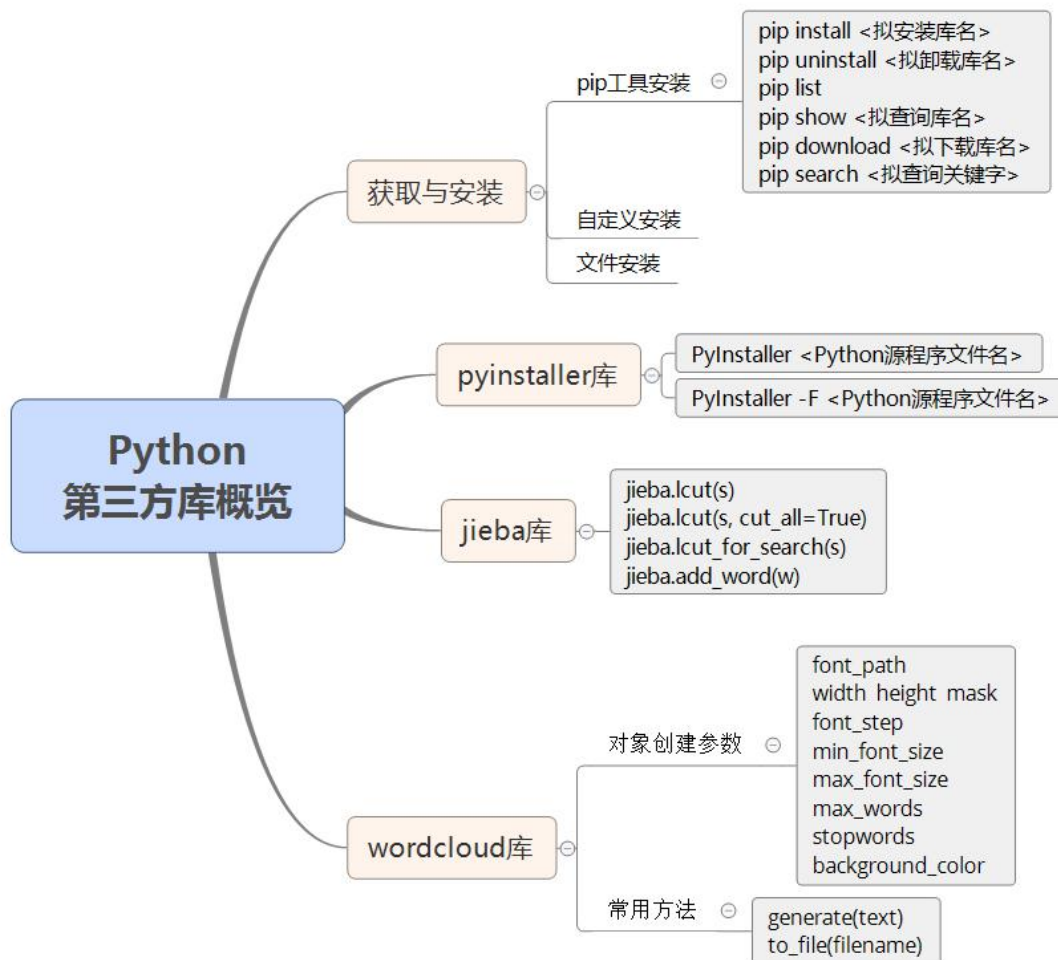
Python第三方库概览



考纲考点

- 第三方库的获取和安装
- 脚本程序转变为可执行程序第三方库：
PyInstaller库(必选)
- 第三方库: jieba库(必选)、wordcloud库 (可选)

知识导图



The Python logo is centered behind the title. It consists of two interlocking snakes, one blue and one yellow, forming a circular shape.

Python第三方库的获取和安装

Python第三方库的获取和安装

- Python第三方库依照安装方式灵活性和难易程度有三个方法：**pip工具安装、自定义安装和文件安装。**



pip工具安装

- 最常用且最高效的Python第三方库安装方式是采用pip工具安装。pip是Python官方提供并维护的在线第三方库安装工具。

pip install <拟安装库名>

```
: \> pip install pygame
...
Installing collected packages: pygame
Successfully installed pygame-1.9.2b1
```



pip工具安装

- pip是Python第三方库最主要的安装方式，可以安装超过90%以上的第三方库。然而，还有一些第三方库无法暂时用pip安装，此时，需要其他的安装方法。
- pip工具与操作系统也有关系，在Mac OS X和Linux等操作系统中，pip工具几乎可以安装任何Python第三方库，在Windows操作系统中，有一些第三方库仍然需要用其他方式尝试安装。

自定义安装

- 自定义安装指按照第三方库提供的步骤和方式安装。第三方库都有主页用于维护库的代码和文档。以科学计算用的numpy为例，开发者维护的官方主页是：

<http://www.numpy.org/>

- 浏览该网页找到下载链接，如下：

<http://www.scipy.org/scipylib/download.html>

- 进而根据指示步骤安装。



文件安装

- 为了解决这类第三方库安装问题，美国加州大学尔湾分校提供了一个页面，帮助Python用户获得Windows可直接安装的第三方库文件，链接地址如下：

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

文件安装

- 这里以scipy为例说明，首先在上述页面中找到scipy库对应的内容。选择其中的.whl文件下载，这里选择适用于Python 3.5版本解释器和32位系统的对应文件：scipy-0.17.1-cp35-cp35m-win32.whl，下载该文件到D:\pycodes目录。
- 然后，采用pip命令安装该文件。

```
:>pip install D:\pycodes\scipy-0.17.1-cp35-cp35m-win32.whl
Processing d:\pycodes\scipy-0.17.1-cp35-cp35m-win32.whl
Installing collected packages: scipy
Successfully installed scipy-0.17.1
```

Python第三方库的获取和安装

- 对于上述三种安装方式，一般**优先选择采用pip工具安装**，如果安装失败，则选择自定义安装或者文件安装。另外，如果需要在没有网络条件下安装Python第三方库，请直接采用文件安装方式。其中，.whl文件可以通过pip download指令在有网络条件的情况下获得。

pip工具使用

■ 执行**pip -h**将列出pip常用的子命令

```
:>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.
  search           Search PyPI for packages.
  wheel            Build wheels from your requirements.
  hash            Compute hashes of package archives.
  completion       A helper command used for command completion
  help            Show help for commands.
```

Python第三方库的获取和安装

- pip支持安装（install）、下载（download）、卸载（uninstall）、列表（list）、查看（list）、查找（search）等一系列安装和维护子命令。

Python第三方库的获取和安装

- pip的uninstall子命令可以卸载一个已经安装的第三方库，格式如下：

`pip uninstall <拟卸载库名>`

- pip的list子命令可以列出当前系统中已经安装的第三方库，格式如下：

`pip list`

Python第三方库的获取和安装

- pip的show子命令列出某个已经安装库的详细信息，格式如下：

```
pip show <拟查询库名>
```

- pip的download子命令可以下载第三方库的安装包，但并不安装，格式如下：

```
pip download
```


Python第三方库的获取和安装

- pip的search子命令可以联网搜索库名或摘要中关键字，格式如下：

```
pip search <拟查询关键字>
```

- 以查询含有installer单词的库为例，执行效果如下：

```
:\>pip search installer
winbrew (1.1.7) - Native package installer for Windows
pygitflow-avh (1.2.0) - Pythonic Installer for Git Flow
(AVH Edition).
notouch (0.3) - Notouch Physical Machine
Installer Automation Service
.....
```

The Python logo, consisting of two interlocking snakes, is centered behind the title text.


PyInstaller库概述



PyInstaller库概述

- PyInstaller是一个十分有用的Python第三方库，它能够在Windows、Linux、Mac OS X等操作系统下将Python源文件打包，**变成直接可运行的可执行文件**。
- 通过对源文件打包，Python程序可以在没有安装Python的环境中运行，也可以作为一个独立文件方便传递和管理。

```
: \> pip install PyInstaller
```

The Python logo is centered behind the title text. It consists of two interlocking snakes, one blue and one yellow, forming a circular shape.

PyInstaller库与程序打包

PyInstaller库与程序打包

- 使用PyInstaller库对Python源文件打包十分简单，使用方法如下：

:\>PyInstaller <Python源程序文件名>

- 执行完毕后，源文件所在目录将生成dist和build两个文件夹。最终的打包程序在dist内部与源文件同名的目录中。

PyInstaller库与程序打包

- 可以通过-F参数对Python源文件生成一个独立的可执行文件，如下：

: \>PyInstaller -F <Python源程序文件名>

```
: \>PyInstaller -F SnowView.py
```

- 执行后在dist目录中出现了SnowView.exe文件，没有任何依赖库，执行它即可显示雪景效果。



PyInstaller库与程序打包

■ PyInstaller有一些常用参数

参数	功能
-h, --help	查看帮助
--clean	清理打包过程中的临时文件
-D, --onedir	默认值，生成dist目录
-F, --onefile	在dist文件夹中只生成独立的打包文件
-i <图标文件名.ico >	指定打包程序使用的图标（icon）文件

The Python logo is centered behind the title. It consists of two interlocking snakes, one blue and one yellow, forming a circular shape.

jieba库概述

jieba库概述

- 由于中文文本中的单词不是通过空格或者标点符号分割，中文及类似语言存在一个重要的“分词”问题。
- jieba（“结巴”）是Python中一个重要的第三方中文分词函数库。

```
: \> pip install jieba
```



jieba库概述

- jieba库的分词原理是利用一个中文词库，将待分词的内容与分词词库进行比对，通过图结构和动态规划方法找到最大概率的词组。除了分词，jieba还提供增加自定义中文单词的功能。

jieba库概述

- jieba库支持三种分词模式：**精确模式**，将句子最精确地切开，适合文本分析；**全模式**，把句子中所有可以成词的词语都扫描出来，速度非常快，但是不能解决歧义；**搜索引擎模式**，在精确模式基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

jieba库概述

- 对中文分词来说，jieba库只需要一行代码即可。

```
>>>import jieba
>>>jieba.lcut("全国计算机等级考试")
Building prefix dict from the default dictionary ...
Loading model from cache C:\AppData\Local\Temp\jieba.cache
Loading model cost 1.001 seconds.
Prefix dict has been built succesfully.
['全国', '计算机', '等级', '考试']
```

The Python logo is centered in the background of the slide. It consists of two interlocking snakes, one blue and one yellow, forming a circular shape. The logo is slightly faded and serves as a backdrop for the title text.

jieba库与中文分词

jieba库与中文分词

- jieba.lcut(s)是最常用的中文分词函数，用于精准模式，即将字符串分割成等量的中文词组，返回结果是列表类型。

```
>>>import jieba  
>>>ls = jieba.lcut("全国计算机等级考试Python科目")  
>>>print(ls)  
['全国', '计算机', '等级', '考试', 'Python', '科目']
```

jieba库与中文分词

- `jieba.lcut(s, cut_all = True)`用于全模式，即将字符串的所有分词可能均列出来，返回结果是列表类型，冗余性最大。

```
>>>import jieba
>>>ls = jieba.lcut("全国计算机等级考试Python科目", cut_all=True)
>>>print(ls)
['全国', '国计', '计算', '计算机', '算机', '等级', '考试',
'Python', '科目']
```

jieba库与中文分词

- `jieba.lcut_for_search(s)`返回搜索引擎模式，该模式首先执行精确模式，然后再对其中长词进一步切分获得最终结果。

```
>>>import jieba
>>>ls = jieba.lcut_for_search("全国计算机等级考试Python科目")
>>>print(ls)
['全国', '计算', '算机', '计算机', '等级', '考试', 'Python', '科目']
```


jieba库与中文分词

- 搜索引擎模式更倾向于寻找短词语，这种方式具有一定冗余度，但冗余度相比全模式较少。
- 如果希望对文本准确分词，不产生冗余，只能选择jieba.lcut(s)函数，即精确模式。如果希望对文本分词更准确，不漏掉任何可能的分词结果，请选用全模式。如果没想好怎么用，可以使用搜索引擎模式。

jieba库与中文分词

- jieba.add_word()函数，顾名思义，用来向jieba词库增加新的单词。

```
>>>import jieba
>>>jieba.add_word("Python科目")
>>>ls = jieba.lcut("全国计算机等级考试Python科目")
>>>print(ls)
['全国', '计算机', '等级', '考试', 'Python科目']
```

The Python logo is centered behind the title. It consists of two interlocking snakes, one blue and one yellow, forming a circular shape.

wordcloud库概述

wordcloud库概述

- 词云以词语为基本单元，根据其在文本中出现的频率设计不同大小以形成视觉上不同效果，形成“关键词云层”或“关键词渲染”，从而使读者只要“一瞥”即可领略文本的主旨。

wordcloud库概述


- wordcloud库是专门用于根据文本生成词云的Python第三方库，十分常用且有趣。
- 装wordcloud库在Windows的cmd命令行使用如下命令：

```
: \> pip install wordcloud
```

wordcloud库概述

- wordcloud库的使用十分简单，以一个字符串为例。其中，产生词云只需要一行语句，在第三行，并可以将词云保存为图片。

```
>>>from wordcloud import WordCloud  
>>>txt='I like python. I am learning python'  
>>>wordcloud = WordCloud().generate(txt)  
>>>wordcloud.to_file('testcloud.png')  
<wordcloud.wordcloud.WordCloud object at 0x000001583E26D208>
```

The Python logo is centered behind the title text. It consists of two interlocking snakes, one blue and one yellow, forming a circular shape.

wordcloud库与可视化词云

wordcloud库与可视化词云

- 在生成词云时，wordcloud默认会以空格或标点为分隔符对目标文本进行分词处理。对于中文文本，分词处理需要由用户来完成。一般步骤是先将文本分词处理，然后以空格拼接，再调用wordcloud库函数。



wordcloud库与可视化词云

```
1 import jieba
2 from wordcloud import WordCloud
3 txt = '程序设计语言是计算机能够理解和识别用户操作意图的一种交互体系，它按
4 照特定规则组织计算机指令，使计算机能够自动进行各种运算处理。'
5
6 words = jieba.lcut(txt)          # 精确分词
7
8 newtxt = ' '.join(words)        # 空格拼接
9
10 wordcloud = WordCloud(font_path="msyh.ttc").generate(newtxt)
11 wordcloud.to_file('词云中文例子图.png')    # 保存图片
```

wordcloud库与可视化词云

- wordcloud库的核心是WordCloud类，所有的功能都封装在WordCloud类中。使用时需要实例化一个WordCloud类的对象，并调用其generate(text)方法将text文本转化为词云。

wordcloud库与可视化词云

■ WordCloud对象创建的常用参数

参数	功能
font_path	指定字体文件的完整路径，默认None
width	生成图片宽度，默认400像素
height	生成图片高度，默认200像素
mask	词云形状，默认None，即，方形图
min_font_size	词云中最小的字体字号，默认4号
font_step	字号步进间隔，默认1
min_font_size	词云中最大的字体字号，默认None，根据高度自动调节
max_words	词云图中最大词数，默认200
stopwords	被排除词列表，排除词不在词云中显示
background_color	图片背景颜色，默认黑色

wordcloud库与可视化词云

■ WordCloud类的常用方法

方法	功能
<code>generate(text)</code>	由text文本生成词云
<code>to_file(filename)</code>	将词云图保存为名为filename的文件

wordcloud库与可视化词云

- 下面以Alice梦游仙境为例，展示参数、方法的使用。



wordcloud库与可视化词云

```
from wordcloud import WordCloud
from scipy.misc import imread

mask = imread('AliceMask.png')

with open('AliceInWonderland.txt', 'r', encoding='utf-8') as file:
    text = file.read()
    wordcloud = WordCloud(background_color="white", \
                           width=800, \
                           height=600, \
                           max_words=200, \
                           max_font_size=80, \
                           mask = mask, \
                           ).generate(text)

# 保存图片
wordcloud.to_file('AliceInWonderland.png')
```

wordcloud库与可视化词云

- 其中，`from scipy.misc import imread`一行用于将AliceMask.png读取为nd-array类型，用于后面传递给mask参数使用。（这个库函数隶属于scipy库，pip在安装wordcloud库时会自动安装依赖库。）



实例解析：《红楼梦》人物出场词云



《红楼梦》人物出场统计

- 《红楼梦》是一本鸿篇巨著，里面出现了几百个各具特色的人物。每次读这本经典作品都会想一个问题，全书这些人物谁出场最多呢？一起来用Python回答这个问题吧。
- 人物出场统计涉及对词汇的统计。中文文章需要分词才能进行词频统计，这需要用到jieba库。

《红楼梦》人物出场统计

```
1  # CalStoryOfStone.py
2  import jieba
3  f = open("红楼梦.txt", "r")
4  txt = f.read()
5  f.close()
6  words = jieba.lcut(t)
7  counts = {}
8  for word in words:
9      if len(word) == 1:  #排除单个字符的分词结果
10         continue
11     else:
12         counts[word] = counts.get(word,0) + 1
13 items = list(counts.items())
14 items.sort(key=lambda x:x[1], reverse=True)
15 for i in range(15):
16     word, count = items[i]
17     print ("{:<10}{1:>5}".format(word, count))
```

《红楼梦》人物出场统计

- 先输出排序前15的单词，运行程序后，输出结果

如下：

```
>>>
宝玉          3748
什么          1613
一个          1451
贾母          1228
我们          1221
那里          1174
凤姐          1100
王夫人        1011
你们          1009
如今          999
说道          973
知道          967
老太太       966
起来          949
姑娘          941
```

《红楼梦》人物出场统计

- 与英文词频统计类似，需要排除一些人名无关词汇，如“什么”、“一个”等。

```
1 # CalStoexcludes = {"什么","一个","我们","那里","你们","如今", \
2                       "说道","知道","老太太","起来","姑娘","这里", \
3                       "出来","他们","众人","自己","一面","太太", \
4                       "只见","怎么","奶奶","两个","没有","不是", \
5                       "不知","这个","听见"}
6 for word in excludes:
7     del(counts[word])
```

《红楼梦》人物出场统计

- 输出排序前5的单词，运行程序后，输出结果如下：

```
>>>
宝玉          3748
贾母          1228
凤姐          1100
王夫人        1011
贾琏          670
```

《红楼梦》人物出场词云

- 结合已经将结果的词云效果，利用wordcloud库，将人物出场统计以词云的方式展现出来
- 使用jieba库进行分词，所不同的是，分词后的结果以空格重新拼接为文本，并由wordcloud进一步处理。无关词汇的排除也可以借助wordcloud中的stopwords参数完成。

《红楼梦》人物出场词云

```

1 import jieba
2 from wordcloud import WordCloud
3
4 excludes = {"什么","一个","我们","那里","你们","如今", \
5             "说道","知道","老太太","起来","姑娘","这里", \
6             "出来","他们","众人","自己","一面","太太", \
7             "只见","怎么","奶奶","两个","没有","不是", \
8             "不知","这个","听见"}
9 f = open("红楼梦.txt", "r")
10 txt = f.read()
11 f.close()
12 words = jieba.lcut(txt)
13 newtxt = ' '.join(words)
14 wordcloud = WordCloud(background_color="white", \
15                        width=800, \
16                        height=600, \
17                        font_path="msyh.ttc", \
18                        max_words=200, \
19                        max_font_size=80, \
20                        stopwords = excludes, \
21                        ).generate(newtxt)
22 wordcloud.to_file('红楼梦基本词云.png')

```

[illegible]

《红楼梦》人物出场词云

- 可以看到，输出结果有很多无关词汇，人物出现并不明显。这说明直接采用分词方式并不能较好达到预期效果。
- 结合对人物出场的前期统计结果，可以将 `max_words=200` 参数改为 `max_words=5`，获得前5个出场次数最多人物组成的词云。

《红楼梦》人物出场词云

贾母 凤姐
宝玉 王夫人

《红楼梦》人物出场词云

- 可以看到，wordcloud库具备基本的统计和排序功能，可以配合分词、整合、排除等功能，合理调整词云设置参数将产生不同的可视化效果，文字过多或过少都不会有太好效果。



本章小结

本章介绍了利用Python第三方库编程的模块编程思想和计算生态的理解和运用，并进一步讲解了如何使用jieba词库对中文文档进行分词并进一步统计文档词频。

本章主要围绕Python第三方库，讲解了第三方库获取和安装方法，并详细介绍了PyInstaller程序打包功能、jieba中文分词功能和wordcloud词云可视化功能等3个具体第三方库的使用。通过《红楼梦》人物出场统计和词云效果展示实例帮助读者熟练掌握这3个Python第三方库的具体使用方法。

古籍中外名著名篇甚多，除了《红楼梦》，还对哪些内容感兴趣？词频统计、人物统计、词云效果，来套组合拳吧！