第 1 章　Python 零基础语法入门

在学习 Python 网络爬虫之前，读者需学习 Python 的基础语法。本章立足基础，讲解 Python 和 PyCharm 的安装及 Python 最简单的语法基础和爬虫技术中所需的 Python 语法。

本章涉及的主要知识点如下。

- Python 和 PyCharm 的安装：学会 Python 和 PyCharm 的安装方法。
- 变量和字符串：学会使用变量和字符串的基本用法。
- 函数与控制语句：学会 Python 循环、判断语句、循环语句和函数的使用。
- Python 数据结构：理解和使用列表、字典、元组和集合。
- Python 文件操作：学习使用 Python 建立文件并写入数据。
- Python 面向对象：了解 Python 中类的定义和使用方法

1.1　Python 与 PyCharm 安装

“工欲善其事，必先利其器”，本节介绍 Python 环境的安装和 Python 的集成开发环境（IDE）PyCharm 的安装。

1.1.1　Python 安装（Windows、Mac 和 Linux）

当前主流的 Python 版本为 2.x 和 3.x。由于 Python 2 第三方库更多（很多库没有向 Python 3 转移），企业普遍使用 Python 2。如果作为学习和研究的话，建议使用 Python 3，因为它是未来的发展方向。所以本教程选择 Python 3 的环境。

1．Windows中安装Python 3

在 Windows 系统中安装 Python 3，请参照下面的步骤进行。

（1）打开浏览器，访问 Python 官网（https://www.python.org/）。

（2）光标移动至 Downloads 链接，单击 Windows 链接。

（3）根据自己的 Windows 版本（32 位或 64 位），下载相应的 Python 3.5 版本，如为 Windows 32 位系统，应下载 Windows x86 executable installer，如果为 Windows 64 位系统，应下载 Windows x86-64 executable installer。

（4）单击运行文件，勾选 Add Python 3.5 to PATH，然后单击 Install Now 按钮即可完成安装。

在计算机中打开命令提示符（cmd）窗口，输入 python，如图 1.1 所示，说明 Python 环境安装成功。

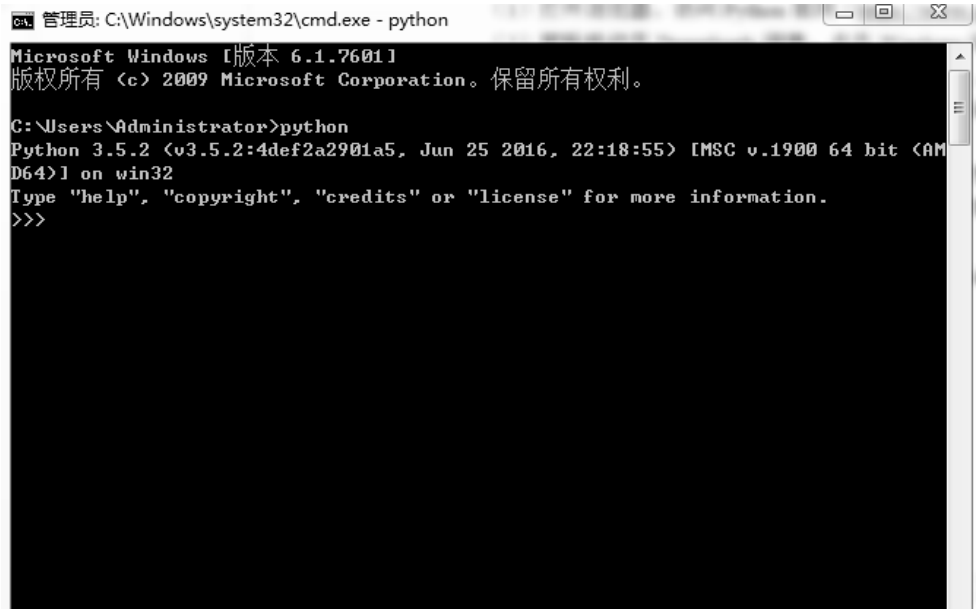

图 1.1　运行 Python 环境

当界面出现提示符>>>时，就表明进入了 Python 交互式环境，输入代码后按 Enter 键即可运行 Python 代码，通过输入 exit()并按 Enter 键，就可以退出 Python 交互式环境。

注意：如果出现错误，可能是因为安装时未勾选 Add Python3.5 to PATH 选项，此时卸载 Python 后重新安装时勾选 Add Python3.5 to PATH 选项即可。

2．Mac中安装Python3

Mac 系统中自带了 Python 2.7，需到 Python 官网上下载并安装 Python 3.5。Mac 系统中的安装比 Windows 更为简单，一直单击“下一步”按钮即可完成。安装完后，打开终端并输入 python3，即可进入 Mac 的 Python 3 的交互式环境。

3．Linux中安装Python 3

大部分 Linux 系统内置了 Python 2 和 Python 3，通过在终端输入 python –version，可以查看当前 Python 3 的版本。如果需要安装某个特定版本的 Python，可以在终端中输入：

```
sudo apt-get install python3.5
```

1.1.2　PyCharm 安装

安装好 Python 环境后，还需要安装一个集成开发环境（IDE），IDE 集成了代码编写功能、分析功能、编译功能和调试功能。在这里向读者推荐一个最智能、好用的 Python IDE，叫做 PyCharm。进入 PyCharm 的官网（http://www.jetbrains.com/pycharm/），下载社区版即可。由于 PyCharm 上手极为简单，因此就不详细讲解 PyCharm 的使用方法了。以下讲解如何使用 PyCharm 关联 Python 解释器，让 PyCharm 可以运行 Python 代码。

（1）打开 PyCharm，在菜单栏中选择 File ｜ Defalut Settings 命令。

（2）在弹出的对话框中选择 Project Interpreter，然后在右边选择 Python 环境，这里选择 Python 3.5，单击 OK 按钮，即可关联 Python 解释器，如图 1.2 所示。

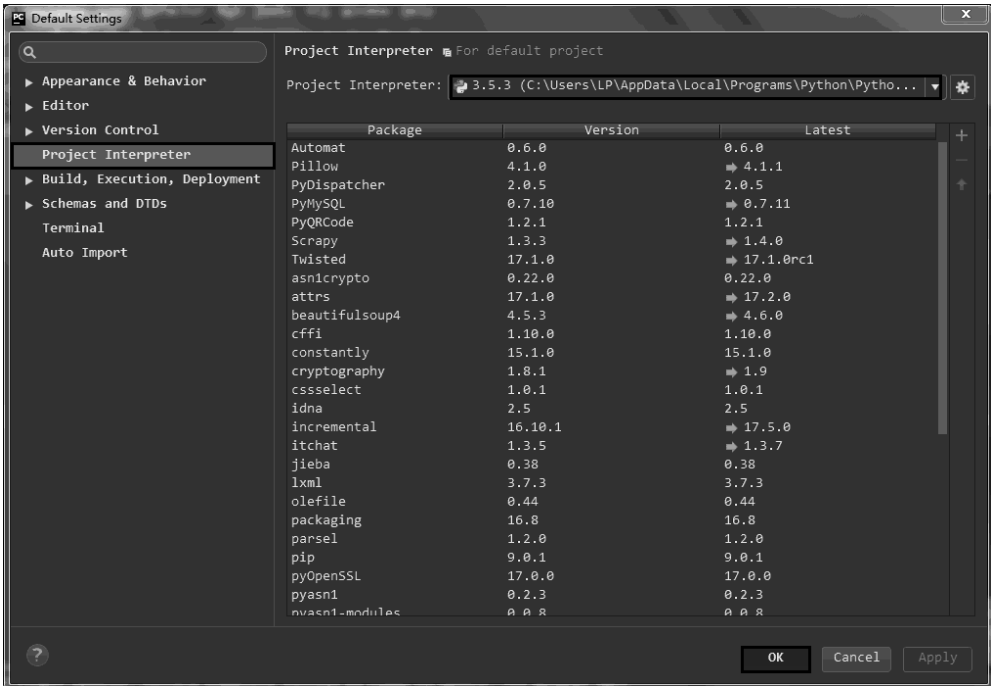

图 1.2　关联 Python 解释器

1.2　变量和字符串

本节主要介绍 Python 变量的概念、字符串的基本使用方法、字符串的切片和索引，以及字符串的几种常用方法。

1.2.1 变量

Python 中的变量很好理解，例如：

```
a = 1
```

这种操作称为赋值，意思为将数值 1 赋给了变量 a。

注意：Python 中语句结束不需要以分号结束，变量不需要提前定义。

现在有变量 a 和变量 b，可以通过下面代码进行变量 a、b 值的对换。

```
a = 4
b = 5
t = a                          #把 a 值赋给 t 变量
a = b                          #把 b 值赋给 a 变量
b = t                          #把 t 值赋给 b 变量
print(a,b)
# result 5 4
```

这种方法类似于将两个杯子中的饮料对换，只需要多加一个杯子，即可完成饮料的对换工作。

1.2.2 字符串的“加法”和“乘法”

由于 Python 爬虫的对象大部分为文本，所以字符串的用法尤为重要。在 Python 中，字符串由双引号或单引号和引号中的字符组成。首先，通过下面代码看看字符串的“加法”：

```
a = 'I'
b = ' love'
c = ' Python'
print(a + b + c)               #字符串相加
# result I love Python
```

在爬虫代码中，会经常构造 URL，例如，在爬取一个网页链接时，只有一部分 /u/9104ebf5e177，这部分链接是无法访问的，还需要 http://www.jianshu.com，这时可以通过字符串的“加法”进行合并。

注意：此网站为笔者的简书首页。

Python 的字符串不仅可以相加，也可以乘以一个数字：

```
a = 'word'
print(a*3)                     #字符串乘法
#result wordwordword
```

字符串乘以一个数字，意思就是将字符串复制这个数字的份数。

1.2.3　字符串的切片和索引

字符串的切片和索引就是通过 string[x]，获取字符串的一部分信息：

```
a = 'I love python'
print(a[0])                    #取字符串第一个元素
#result I
print(a[0:5])                  #取字符串第一个到第五个元素
#result I lov
print(a[-1])                   #取字符串最后一个元素
#result n
```

通过图 1.3 就能清楚地理解字符串的切片和索引。

I		l	o	v	e		p	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

图 1.3　字符串切片和索引

注意：a[0:5]中的第 5 个是不会选择的。

在爬虫实战中，经常会通过字符串的切片和索引，提取需要的部分，剔除一些不需要的部分。

1.2.4　字符串方法

Python 作为面向对象的语言，每个对象都有相应的方法，字符串也一样，拥有多种方法，在这里介绍爬虫技术中常用的几种方法。

1．split()方法

```
a = 'www.baidu.com'
print(a.split('.'))
# result ['www', 'baidu', 'com']
```

字符串的 split()方法就是通过给定的分隔符（在这里为‘.’），将一个字符串分割为一个列表（后面将详细讲解列表）。

注意：如果没有提供任何分隔符，程序会把所有的空格作为分隔符（空格、制表、换行等）。

2．repalce()方法

```
a = 'There is apples'
b = a.replace('is','are')
```

```
print(b)
# result There are apples
```

这种方法类似文本中的“查找和替换”功能。

3．strip()方法

```
a = '  python is cool    '
print(a.strip())
# result  python is cool
```

strip()方法返回去除两侧（不包括内部）空格的字符串，也可以指定需要去除的字符，将它们列为参数中即可。

```
a = '***python *is *good***'
print(a.strip('*!'))
# result  python *is *good
```

这个方法只能去除两侧的字符，在爬虫得到的文本中，文本两侧常会有多余的空格，只需使用字符串的 strip()方法即可去除多余的空格部分。

4．format()方法

最后，再讲解下好用的字符串格式化符，首先看以下代码：

```
a = '{} is my love'.format('Python')
print(a)
# result  Python is my love
```

字符串格式化符就像是做选择题，留了空给做题者选择。在爬虫过程中，有些网页链接的部分参数是可变的，这时使用字符串格式化符可以减少代码的使用量。例如，Pexels 素材网（https://www.pexels.com/），当搜索图片时，网页链接也会发生变化，如在搜索栏中输入 book，网页跳转为 https://www.pexels.com/search/book/，可以设计如下代码，笔者只需输入搜索内容，便可返回网页链接。

```
content = input('请输入搜索内容：')
url_path = 'https://www.pexels.com/search/{}/'.format(content)
print(url_path)
```

运行程序并输入 book，便可返回网页链接，单击网页链接便可访问网页了，如图 1.4 所示。

图 1.4　字符串格式化符演示

注意：Pexels 素材网为外文网，需输入英文，该网站图片免费下载，无须担忧版权问题。