第 1 章
Python 基础

1.1 Python简介与安装

1.1.1 Python 简介

全世界有 600 多种编程语言，流行的编程语言有 20 余种。你可能已经听说过很多种流行的编程语言，比如 C 语言、C++、Java、Visual Basic、JavaScript 等。Python 也是一种计算机程序设计语言，是著名的“龟叔”Guido van Rossum 在 1989 年圣诞节期间，为了打发无聊的圣诞节而编写的一种编程语言。

目前，Python 已经成为最受欢迎的程序设计语言之一。2011 年 1 月，它被 TIOBE 编程语言排行榜评为“2010 年度语言”。自从 2004 年以后，Python 的使用率呈线性增长。

Python 语言是少有的一种可以称得上既简单又功能强大的编程语言。使用 Python 语言后你会惊喜地发现编程是多么简单，它注重的是如何解决问题而不是编程语言的语法和结构。

由于 Python 语言的简洁性、易读性以及可扩展性，在国外用 Python 做科学计算的研究机构日益增多，一些知名大学已经采用 Python 来教授程序设计课程。例如，卡耐基梅隆大学的编程基础、麻省理工学院的计算机科学及编程导论就使用 Python 语言讲授。许多大型网站也是用 Python 开发的，例如，YouTube、Instagram 以及豆瓣。很多大公司，包括 Google、Yahoo 等，甚至 NASA（美国航空航天局）都大量地使用 Python。

众多开源的科学计算软件包都提供了 Python 的调用接口，例如，著名的计算机视觉库 OpenCV、三维可视化库 VTK、医学图像处理库 ITK。而 Python 专用的科学计算扩展库就更多了，例如，这 3 个经典的科学计算扩展库 NumPy、SciPy 和 Matplotlib，它们分别为 Python 提供了快速数组处理、数值运算以及绘图功能。因此 Python 语言及其众多的扩展库

Ypwzw.indd 1 2019.5.13 5:15:25 PM

所构成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表，甚至开发科学计算应用程序。

Python 适合开发的应用，首先是网络应用，包括网站、后台服务等；其次是许多日常需要的小工具，包括系统管理员需要的脚本任务等；另外，就是把其他语言开发的程序再包装起来，方便使用。本书主要介绍 Python 在智能机器人方面的应用。

Python 是开源的、免费的，网上有大量的资源，下面几个重要网址推荐给你使用，里面包含了大量的开发资源和源代码。

（1）https://www.python.org/

（2）https://www.python.org/doc/

（3）http://bugs.python.org/

（4）https://hackerone.com/python

（5）http://stackoverflow.com/questions/tagged/python

1.1.2 Python 的特点

Python 的设计哲学是“优雅”“明确”“简单”，即尽量写容易看明白的代码，尽量少写代码。

任何编程语言都有优缺点，Python 也不例外。Python 语言的优点如下。

（1）非常简单，非常适合阅读。阅读一个良好的 Python 程序感觉就像是在读英语，尽管这个“英语”的语法要求非常严格！ Python 的这种伪代码本质是它最大的优点。它使你专注于解决问题而不是去搞明白语言本身。

（2）易学。Python 虽然是用 C 语言写的，但它摈弃了 C 语言中非常复杂的指针，简化了 Python 的语法。

（3）Python 是 FLOSS（自由 / 开放源码软件）之一。简单地说，你可以自由地发布这个软件的拷贝，阅读它的源代码，对它做改动，把它的一部分用于新的自由软件中。

（4）可移植性。由于它的开源本质，Python 已经被移植在许多平台上（经过改动使它能够工作在不同平台上）。如果你已经小心地避免使用 Python 中依赖于系统的特性，那么你的所有 Python 程序无须修改就可以在大部分的主流系统平台上运行。这些平台包括 Linux、Windows、FreeBSD、Macintosh、Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acom RISC OS、VxWorks、PlayStation、Sharp Zaurus、Windows CE，甚至还有 PocketPC、Symbian 以及 Google 基于 Linux 开发的 Android 平台。

（5）在计算机内部，Python 解释器把源代码转换成称为字节码的中间形式，然后再把它翻译成计算机使用的机器语言并运行。由于不再担心如何编译程序，如何确保连接转载正确的库等，这使得人们使用 Python 更加简单，只需要把 Python 程序复制到另外一台计算机上，它就可以工作了，这也使 Python 程序更加易于移植。

（6）Python 既支持面向过程的函数编程，也支持面向对象的抽象编程。在面向过程的语言中，程序是由过程或仅仅是可重用代码的函数构建起来的。在面向对象的语言中，程序是由数据和功能组合而成的对象构建起来的。与其他主要的语言如 C++ 和 Java 相比，Python 以一种非常强大又简单的方式实现面向对象编程。

（7）可扩展性和可嵌入性。如果需要自己的一段关键代码运行得更快或者希望某些算法不公开，你可以把这部分程序用 C 或 C++ 编写，然后在 Python 程序中使用它们。还可以把 Python 嵌入 C 或 C++ 程序，从而向程序用户提供脚本功能。

（8）丰富的库。Python 标准库很庞大，还有可定义的第三方库可以使用，它可以帮助你处理各种工作，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV 文件、密码系统、GUI（图形用户界面）、Tk 和其他与系统有关的操作。只要安装了 Python，所有这些功能都是可用的，这被称作 Python 的“功能齐全”理念。除了标准库以外，还有许多其他高质量的库，如 wxPython、Twisted 和 Python 图像库等。

（9）规范的代码。Python 采用强制缩进的方式使代码具有极佳的可读性。

Python 语言的缺点如下。

（1）运行速度慢。Python 是解释性语言。若有速度要求，可以用 C++ 改写关键部分。不过对于用户而言，机器上运行速度是可以忽略的，因为用户根本感觉不出这种速度的差异。

（2）单行语句和命令行输出问题：很多时候不能将程序连写成一行。

（3）Python 的开源性使 Python 语言不能加密。

（4）构架选择太多（没有像 C# 这样的官方 .net 构架，也没有像 Ruby 构架开发的相对集中）。不过这也从另一方面说明，Python 比较优秀，吸引的人才多，项目也多。

1.1.3　Windows 下 Python 开发环境搭建

Python 的开发环境很人性化，没有过多烦琐的配置，下载以后直接安装就可以使用，其操作步骤如下。

（1）下载一个 Python IDLE 程序安装包。

Ypwzw.indd 3　2019.5.13 5:15:25 PM

打开 Python 的官方网站 www.python.org，单击 Downloads 按钮，进入页面后，找到适合自己系统的版本安装包，注意 32 位和 64 位安装包的区别。

（2）找到刚下载的 Python 程序安装包，双击打开，运行安装程序。一般无须过多设置，直接单击“下一步”按钮，直至安装成功，单击“完成”按钮就可以了。

（3）从开始菜单找到 Python IDLE，即可启动 Python 集成化编程环境，并可以看到当前安装的 Python 的版本号，如图 1-1 所示。

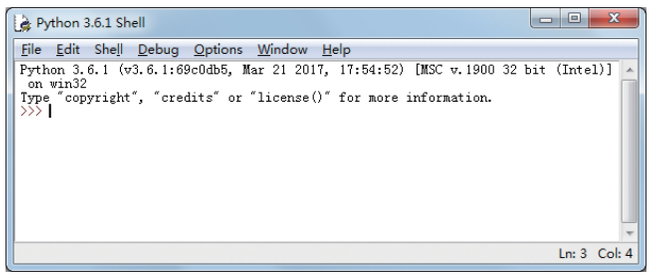

图1-1　当前安装的Python的版本号

除了默认安装的 IDLE，还有大量其他的 IDE 开发环境，例如 Wing IDE、PyCharm、PythonWin、Spyder 等。每个 IDE 都有不同的风格，也分别得到了不同开发人员的喜爱，但严格来说，这些开发环境都是对 Python 解释器 python.exe 的封装，核心是一样的，只是加了一个“外挂”而已，使用起来更加方便，减少了出错率，尤其是拼写错误。本书 Windows 环境下的 Python 开发环境选用官方标准开发环境 IDLE。

1.1.4　模块安装

Python 之所以得到青睐，与涉及各行业、各领域开发的扩展库有很大关系，它的扩展库不仅数量众多、功能强大，而且用起来很方便。虽然 Python 标准库已经提供了非常强大的功能，但很多时候若能够熟练运用扩展库，会事半功倍，大幅提高开发速度。

可以把 Python 模块看作是一个个用来存放积木的收纳箱，每个收纳箱里放着很多特定类型的积木（函数或类），用的时候，我们把收纳箱从仓库里拖出来，然后打开它，选择合适的积木搭建自己的房子、汽车、轮船、飞机等作品（程序）就可以了。

目前，pip 已经成为管理 Python 扩展库的主流方式。使用 pip 不仅可以实时查看本机已安装的 Python 扩展库列表，还支持 Python 扩展库的安装、升级和卸载等操作。使用 pip 工具管理 Python 扩展库只需要保证计算机联网的情况下，输入几个命令即可完成，极大地方便了用户。常用的 pip 命令使用方法如表 1-1 所示。

Ypwzw.indd 4　2019.5.13 5:15:25 PM

表 1-1　常用 pip 命令的使用方法

pip 命令示例	说　明
pip freeze [> requirements.txt]	以 requirements 的格式列出已安装模块
pip list	列出当前已安装的所有模块
pip install SomePackage[==version]	在线安装 SomePackage 模块的指定版本
pip install SomePackage.whl	通过 whl 文件离线安装扩展库
pip install package1 package2...	依次（在线）安装 package1、package2 等扩展模块
pip install -r requirements.txt	安装 requirements.txt 文件中指定的扩展库
pip install -upgrade SomePackage	升级 SomePackage 模块
pip uninstall SomePackage[==version]	卸载 SomePackage 模块的指定版本

使用 pip 命令安装 Python 扩展库需要在命令提示符环境中进行，并且需要切换至 pip 命令所在的目录。

Python 默认安装仅包含部分基本或核心模块，启动时也仅加载了基本模块，在需要的时候再加载其他模块，这样可以减小程序运行的压力，且具有很强的可扩展性，有助于提高系统的安全性。

安装完毕的模块，使用 import 命令导入，之后就可以使用模块中的函数或类了。

1.1.5　IDLE 简单应用

启动 IDLE 之后默认为交互模式，直接在 Python 提示符“>>>”后面输入相应的命令并按 Enter 键执行即可。如果顺利，马上就可以看到执行结果，否则会显示异常。例如：

```
>>> 3+5                          # 计算 3+5 的和
8
>>> 3*(2+6)                      # 计算表达式的值
24
>>>2**99                        # 计算 2 的 99 次方
633825300114114700748351602688
>>> print（"hello world"）# 输出显示 "hello world"
hello world

>>> 2/0                          # 除数为 0，抛出异常
Traceback (most recent call last):
   File "<pyshell#18>", line 1, in <module>
     2/0
ZeroDivisionError: integer division or modulo by zero
>>>import math                  # 导入 math 模块
```

```
>>>math.sin(0.5)              # 求 0.5 弧度的正弦值
0.479425538604203
```

在 IDLE 界面中，选择菜单命令 File→New File，进入程序编辑模式，创建一个程序文件，输入代码并保存为 .py 或 .pyw 文件。

使用菜单命令 Run → Check Module 检查程序中是否存在语法错误，或者使用菜单命令 Run → Run Module 运行程序，程序运行结果将直接显示在 IDLE 交互界面上。

1.1.6 Python 文件名

.py：Python 源文件，由 Python 解释器负责解释执行。

.pyw：Python 源文件，常用于图形界面程序文件。

.pyc：Python 字节码文件，无法使用文本编辑器直接查看该类型文件内容，可用于隐藏 Python 源代码和提高运行速度。

Python 模块第一次被导入时，将被编译成字节码的形式，并在以后再次导入时优先使用 .pyc 文件，以提高模块的加载和运行速度。

1.2 Python基本语法

1.2.1 Python 的对象模型

对象是 Python 语言中最基本的概念，在 Python 中处理的一切都是对象。Python 中有许多内置对象可供编程者使用，内置对象可直接使用，如数字、字符串、列表、del 等；非内置对象需要导入模块才能使用，如要使用正弦函数 sin( )，需要导入 math 模块；要使用随机数产生函数 random( )，需要导入 random 模块等。常见的 Python 对象如表 1-2 所示。

表 1-2 常见的 Python 对象

对象类型	类型名称	示　例	简 要 说 明
数字	int, float, complex	`1234,3.14, 1.3e5, 3+4j`	数字大小没有限制，内置，支持复数及其运算
字符串	str	`'swfu', "I'm student", '''Python ''', r'abc', R'bcd'`	使用单引号、双引号、三引号作为定界符，以字母 r 或 R 引导的表示原始字符串
字节串	bytes	`b'hello world'`	以字母 b 引导，可以使用单引号、双引号、三引号作为定界符

续表

对象类型	类型名称	示　　例	简 要 说 明
列表	list	`[1,2,3],['a', 'b', ['c', 2]]`	所有元素放在一对方括号中，元素之间使用逗号分隔，其中的元素可以是任意类型
字典	dict	`{1:'food',2:'taste',3:'import'}`	所有元素放在一对大括号中，元素之间使用逗号分隔，元素形式为“键 : 值”
元组	tuple	`(2, -5, 6), (3,)`	所有元素放在一对圆括号中，元素之间使用逗号分隔，如果元组中只有一个元素，后面的逗号不能省略

1.2.2　Python 的注释

一个好的、可读性强的程序一般包含 30% 以上的注释。常用的注释方式主要有以下两种。

（1）以 # 开始，表示本行 # 之后的内容为注释。

例如：

```
# 这是一个注释
print("Hello, World!")
```

（2）包含在一对三引号 '''...''' 或 """...""" 之间且不属于任何语句的内容将被解释器认为是注释。

例如：

```
'''
这是多行注释，用三个单引号
这是多行注释，用三个单引号
这是多行注释，用三个单引号
'''
print("Hello, World!")
```

1.2.3　Python 变量

在 Python 中，不需要事先声明变量名及其类型，直接赋值即可创建各种类型的对象变量。这一点适用于 Python 任意类型的对象。

例如：

```
>>> x = 3        # 创建了整型变量 x，并赋值为 3
```

图1-2　变量创建示意图

变量创建示意如图 1-2 所示。

Ypwzw.indd 7　　2019.5.13 5:15:26 PM

```
>>> x = 'Hello world.'    # 创建了字符串变量 x，并赋值为 'Hello world.'
```

Python 属于强类型编程语言，Python 解释器会根据赋值或运算自动推断变量类型。

Python 还是一种动态类型语言，变量的类型也可以随时变化。

```
>>> x = 3
>>> print(type(x))          # 查看变量类型
<class'int'>                # 整数类型

>>> x = 'Hello world.'
>>> print(type(x))          # 查看变量类型
<class'str'>                # 字符串类型

>>> x = [1,2,3]
>>> print(type(x))          # 查看变量类型
<class'list'>               # 列表类型

>>> isinstance(3, int)      # 测试对象是否是某个类型的实例
True

>>> isinstance('Hello world', str)
True
```


在 Python 中，允许多个变量指向同一个值，例如：

```
>>> x = 3
>>> id(x)
1786684560                  # 变量 x 的内存地址

>>> y = x
>>> id(y)                   # 变量 y 的内存地址
1786684560
```

然而，当为其中一个变量修改值以后，其内存地址将会变化，但这并不影响另一个变量，例如，接着上面的代码继续执行下面的代码：

```
>>> x += 6
>>> id(x)
1786684752

>>> y
3
>>> id(y)
1786684560
```

Ypwzw.indd 8 2019.5.13 5:15:27 PM

Python 采用的是基于值的内存管理方式，如果为不同变量赋值为相同值，这个值在内存中只有一份，多个变量指向同一块内存地址，如图 1-3 所示。

例如：

```
>>> x = 3
>>> id(x)
10417624

>>> y = 3
>>> id(y)
10417624
>>> x = [1, 1, 1, 1]
>>> id(x[0]) == id(x[1])
True
```

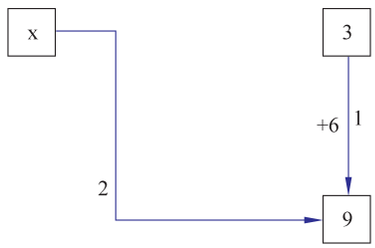

图1-3　内在地址示意图

Python 具有自动内存管理功能，对于没有任何变量指向的值，Python 自动将其删除。因为 Python 会跟踪所有的值，并自动删除不再有变量指向的值。所以 Python 程序员一般情况下不需要太多考虑内存管理的问题。

在定义变量名时，需要注意以下问题。

（1）变量名必须以字母或下划线开头，但以下划线开头的变量在 Python 中有特殊含义。

（2）变量名中不能有空格以及标点符号（括号、引号、逗号、斜线、反斜线、冒号、句号、问号等）。

（3）不能使用关键字作变量名，可以导入 keyword 模块后使用 print(keyword.kwlist) 查看所有 Python 的关键字。

（4）不建议使用系统内置的模块名、类型名或函数名以及已导入的模块名及其成员名作变量名，这将会改变其类型和含义，可以通过 dir(_ _builtins_ _) 查看所有内置模块、类型和函数。

（5）变量名对英文字母的大小写敏感，例如，student 和 Student 是不同的变量。

1.2.4　Python 数字

数字是不可变对象，可以表示任意大小的数字。例如：

```
>>> a=99999999999999999999999999999999
>>> a*a
9999999999999999999999999999999980000000000000000000000000000001
>>> a**3    #a 的三次方
```

Ypwzw.indd 9　　2019.5.13 5:15:29 PM

```
    99999999999999999999999999999997000000000000000000000000000000002
999999999999999999999999999999999
```

Python 的 IDEL 交互界面可以当作简便计算器使用。例如：

```
>>> ((3**2) + (4**2)) ** 0.5
5.0
```

Python 中的整数类型可以分为如下几种。

（1）十进制整数，如 0、–1、9、123。

（2）十六进制整数，需要 16 个数字或字符 0、1、2、3、4、5、6、7、8、9、a、b、c、d、e、f 表示整数，必须以 0x 开头，如 0x10、0xfa、0xabcdef。

（3）八进制整数，只需要 8 个数字 0、1、2、3、4、5、6、7 表示整数，必须以 0o 开头，如 0o35、0o11。

（4）二进制整数，只需要 2 个数字 0、1 表示整数，必须以 0b 开头，如 0b101、0b100。

（5）浮点数又称小数，如 15.0、0.37、–11.2、1.2e2、314.15e–2。

（6）Python 内置支持复数类型，例如：

```
>>> a = 3+4j
>>> b = 5+6j
>>> c = a+b
>>> c
(8+10j)
>>> c.real                  # 查看复数实部
8.0
>>> c.imag                  # 查看复数虚部
10.0
>>> a.conjugate()           # 返回共轭复数
(3-4j)
>>> a*b                     # 复数乘法
(-9+38j)
>>> a/b                     # 复数除法
(0.6393442622950819+0.03278688524590165j)
```

1.2.5 Python 字符串

用单引号、双引号或三引号括起来的符号系列称为字符串。

单引号、双引号、三单引号、三双引号可以互相嵌套，用来表示复杂字符串，例如：'abc'、'123'、' 中国 '、"Python"、'"Tom said, "Let's go""' 都是符合语法的字符串。

Ypwzw.indd 10 2019.5.13 5:15:29 PM