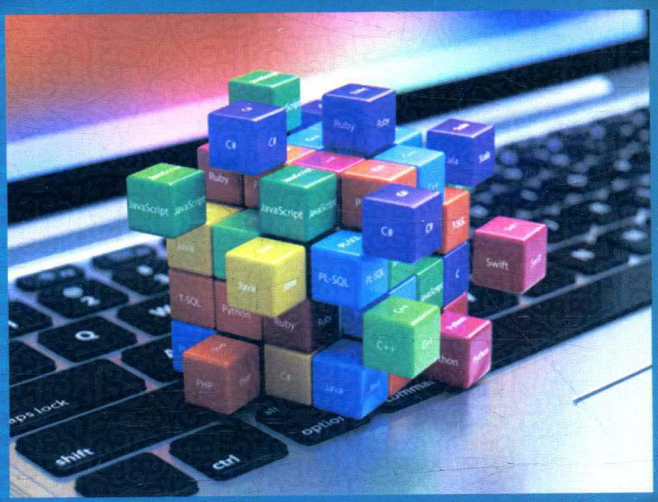
“十三五”普通高等教育规划教材

Python程序设计

从编程基础到专业应用

章 宁 李海峰 主编



提供电子课件、源代码

http://www.cmpedu.com

机械工业出版社
CHINA MACHINE PRESS

“十三五”普通高等教育规划教材

Python 程序设计

——从编程基础到专业应用

章　宁　李海峰　主　编

王　悦　刘灿涛　郭韦昱　副主编

机械工业出版社

本书包含编程基础和专业应用两个部分：基础部分突出 Python 易上手的语法特点，初步培养学生的编程思维；专业应用部分突出 Python 擅长处理数据的特点，设置不同的应用模块，供不同专业选择使用。本书支持任务驱动的教学理念：每章章首给出本章要完成的任务，而且所有知识点均围绕任务实现；全书两个部分的内容分别用两个完整案例贯穿。

本书面向高等院校非计算机类专业的各年级本科生，可作为入门编程课程的教材。

本书配套授课电子课件，需要的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取（QQ：2850823885。电话：010-88379739）。

图书在版编目（CIP）数据

Python 程序设计：从编程基础到专业应用 / 章宁，李海峰主编. —北京：机械工业出版社，2019.1
“十三五”普通高等教育规划教材

ISBN 978-7-111-62013-6

Ⅰ. ①P… Ⅱ. ①章… ②李… Ⅲ. ①软件工具－程序设计－高等学校－教材 Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字（2019）第 028904 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）

策划编辑：郝建伟　　责任编辑：郝建伟
责任校对：张艳霞　　责任印制：李　昂

河北鹏盛贤印刷有限公司印刷

2019 年 3 月第 1 版 • 第 1 次印刷

184mm×260mm • 12.5 印张 • 303 千字

0001－3000 册

标准书号：ISBN 978-7-111-62013-6
定价：42.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
服务咨询热线：（010）88379833
读者购书热线：（010）88379649

网络服务
机 工 官 网：www.cmpbook.com
机 工 官 博：weibo.com/cmp1952
教育服务网：www.cmpedu.com
金　书　网：www.golden-book.com

前　言

本书面向高等院校非计算机类专业的各年级本科生，作为入门编程课程的教材。随着大数据时代的到来，编程语言的应用已经成为各专业所需，目前有不少高校已经面向全校各专业各年级开设《Python 程序设计》公共选修课。

在编程语言的世界里，Python 在近些年获得了最多的关注和发展，成为美国大学最受欢迎的程序设计语言。以美国斯坦福大学为例，该校从 2009 年开设 Python，之后每年新增 5 门左右与 Python 相关的课程，截止到 2015 年，该校共开设 22 门与 Python 相关的课程，并替换了之前的 Java 语言和部分专业的 C 语言课程。Python 是目前最为灵活、最接近自然语言的通用编程语言，功能强大，适合解决各类计算问题。Python 轻语法重应用的特性使得它非常容易上手，有助于初学者形成良好的编程习惯和思维，对于非计算机专业的本科生来说，Python 无疑成为开设程序设计语言课程的首选。

本书深入浅出，充分发挥 Python 语言易上手和擅长数据处理的特点，内容上既包括 Python 编程基础，又包括 Python 专业应用。一方面通过 Python 易上手的特点帮助学生构建良好的编程思维，另一方面通过 Python 擅长数据处理的特点帮助学生在自己的专业领域内形成自主学习应用 Python 的能力。全书内容共 10 章，分别是：Python 起步、基本数据类型、程序的控制结构、组合数据类型、函数、类、模块、文件和异常处理、网络数据爬取和数据可视化。本书采用任务驱动的教学理念，每章章首给出本章要完成的任务，所有知识点均围绕任务实现；编程基础和专业应用分别用两个完整案例贯穿，1～8 章的贯穿案例是“在程序的世界里看见自己”，9～10 章的贯穿案例是“在商务世界里看见数据”。本教材适用于 32～48 课时的弹性教学。1～8 章内容的教学大约需要 24 课时，9～10 章内容的教学则可根据专业需要进行选择，每章 8～12 课时。

本书由中央财经大学信息学院软件开发课程群教学团队编写完成，该团队成员具有多年程序设计和软件开发教学和实践经验，主持完成的教学成果“跨学科的软件开发课程群平台搭建及资源共享机制研究与实践”获 2013 年北京市高等教育教学成果二等奖。团队负责人章宁教授还主持有国家双语教学示范课程，采用全英文方式为计算机及其相关专业的本科生讲授 Python 语言及其应用。章宁教授负责本书的整体策划和最终

统稿，并编写第 1、2 章；信息管理系李海峰副教授编写第 7、8、10 章，王悦副教授编写第 3、6 章；计算机系刘灿涛博士编写第 4、5 章，郭韦昱博士编写第 9 章。柳慧同学参与了全书的编写、校对及统稿工作，秦思佳同学参与了第 7、8、10 章的编写工作，姚苏芮同学参与了第 3 章的编写工作，刘嘉庆同学参与了第 6 章的编写工作。

由于作者水平有限，加之本书编写内容涉及广泛，且信息技术的发展日新月异，因此书中难免存在不妥之处，诚望读者批评赐教，为培养高等院校各年级各专业本科生的编程思维和专业应用能力共同努力。

编　者

目　录

第 1 章　Python 起步

本章将学习 Python 作为一种编程语言的特性，如何安装 Python 开发环境，如何运行 Python 程序，以及编写程序的一些基础知识，包括什么是数据类型、什么是变量、什么是函数、什么是语句等。在章首案例的指引下，本章将介绍如何创建你的第一个 Python 程序文件并运行它。

1.1　案例：在程序的世界里看见自己——世界你好

在本章案例中，将在编程语言的世界里和大家打个招呼。首先在用户终端输入自己的姓（last name）和名（first name），然后通过屏幕和大家打招呼，告诉大家你的姓名，如图 1-1 所示，前两行是输入，第三行是输出。

```
Please input your last name: Li
Please input your first name: Ming
Hello World! My last name is Li and my first name is Ming.
```

图 1-1　案例：世界你好

1.2　Python 的起源和特性

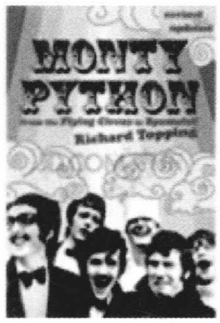
本节首先介绍 Python 的起源，然后介绍其作为一种编程语言的重要特性，最后对各类编程语言的流行程度进行比较。

1.2.1　Python 的起源

Python 的始创者是荷兰人 Guido von Rossum（图 1-2a）。1982 年，Guido 从阿姆斯特丹大学获得了数学和计算机硕士学位。1989 年圣诞节期间，在阿姆斯特丹，Guido 为了打发圣诞节的无趣，决心开发一个新的脚本解释程序。Python 的第一个公开发行版发行于 1991 年。Python 这一称号来自英国肥皂剧《Monty Python》（图 1-2b），之所以当初 Guido 用 Python 作为语言的名字，是因为他太喜欢这部肥皂剧了。

2000 年，Python 2.0 的正式发布，开启了其被广泛应用的新时代。2008 年，Python 3.0 正式发布，但是 3.0 版本无法向下兼容 2.0 版本的既有语法。2010 年，Python 2.x 系列发布了最后一版，即 2.7 版，从此终结了 2.x 系列版本的发展。

a)



b)

图 1-2　Guido 和他喜欢的肥皂剧

a) Guido von Rossum　b) 肥皂剧《Monty Python》

1.2.2　Python 的特性

除 Python 语言外，常用的编程语言还包括 C、C++、C#、Java、JavaScript、R、PHP、Ruby、Matlab、HTML、Perl、Fortran、谷歌的 Go 语言和苹果的 Swift 语言等。Python 是目前最为灵活、最接近自然语言的通用编程语言，它功能强大，适合解决各类计算问题，特别是数据获取和分析处理。

1．高级语言

编程语言包括机器语言、汇编语言和高级语言三大类。机器语言使用二进制代码表达指令，是计算机硬件可以直接识别和执行的编程语言，用机器语言编写程序十分烦冗，程序也难以阅读和修改。汇编语言使用助记符与机器语言中的指令进行一一对应，在计算机发展早期能够帮助程序员提高编程效率。汇编语言和机器语言统称为低级语言。高级语言是接近自然语言的编程语言，可以更容易地描述计算问题并利用计算机解决计算问题。第一个被广泛应用的高级语言是诞生于 1972 年的 C 语言，之后的 40 多年先后诞生了 600 多种编程语言，其中大多数语言由于应用领域狭窄而退出了历史舞台。

2．通用编程语言

通用编程语言是指能够用于编写多种用途程序的编程语言，语法中没有专门用于特定应用的程序元素，如 Python、C、C++、C#、Java 等。专用编程语言是指包含针对特定应用的程序元素，或者应用领域比较狭窄的编程语言，如 HTML、JavaScript、Matlab、PHP 等。Python 可以用于编写各个领域的应用程序，从科学计算、数据处理到人工智能、机器人、区块链，Python 语言都能够发挥重要作用。

3．动态语言

高级语言按照计算机执行方式的不同可以分成两类：采用解释执行的动态语言和采

用编译执行的静态语言。编译是将源代码（高级语言代码）转换成目标代码（机器语言代码）的过程，执行编译的计算机程序称为编译器（Compiler）。解释是将源代码逐条转换成目标代码同时逐条运行目标代码的过程，执行解释的计算机程序称为解释器（Interpreter）。解释和编译的区别在于：编译是一次性地翻译，一旦程序被编译，就不再需要编译器或者源代码；解释则在每次程序运行时都需要解释器和源代码。这两者的区别类似于外语资料的翻译和实时的同声传译。C、Java 是采用编译执行的静态语言，而 JavaScript、PHP 则是采用解释执行的动态语言。Python 语言是采用解释执行方式的现代动态语言，其解释器保留了编译器的部分功能，随着程序运行，解释器也会生成一个完整的目标代码，从而提升了计算机性能。

4．开源

开源（Open Source）指的是开放源代码，即源代码公开，任何人都可以访问、学习、修改甚至发布。Python 语言是开源项目的优秀代表，其解释器的全部代码都是开放的，任何计算机高手都可以为不断推动 Python 语言的发展做出贡献。Python 软件基金会（Python Software Foundation）作为一个非营利组织，拥有 Python 2.1 版本之后所有版本的版权，该组织致力于更好地推进并保护 Python 语言的开放性。世界各地的程序员通过开源社区贡献了十几万个第三方函数库，几乎覆盖了计算机技术的每个领域。

5．语法特点

1）简洁易学：Python 语言关键字少、结构简单、语法清晰，实现相同功能的代码行数仅为其他语言的 1/10～1/5，初学者可以在短时间内轻松上手。

2）强制可读：Python 语言通过强制缩进（类似文章段落的首行空格）来体现语句间的逻辑关系，显著提高了程序的可读性。

3）支持中文：Python 3.0 解释器采用 UTF-8 编码表达所有字符，可以表达英文、中文、韩文、法文等各类自然语言。

1.2.3　各类编程语言的比较

从 2014 年开始，IEEE Spectrum 杂志每年都会发布编程语言排行榜，2014～2017 年的榜单如图 1-3 所示。

Python 的排名从 2016 年开始就借助人工智能的发展持续上升，到 2017 年已经成为第一名。始终排在前四的 Python、C、Java 和 C++都拥有广大的用户群体，且用户总量也十分接近。C#、R 也是非常流行的编程语言。当然排行榜仅供参考，因为评价参数及权重不同，会导致不同的排行结果。

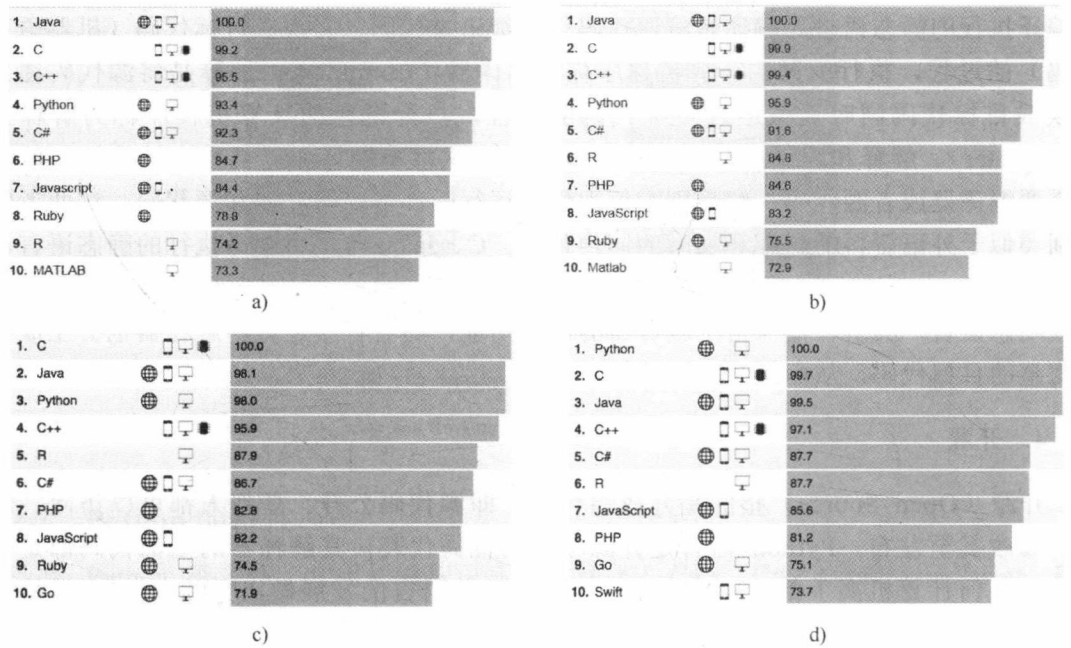


图 1-3　编程语言排行

a) 2014 年　b) 2015 年　c) 2016 年　d) 2017 年

1.3　安装和运行

本节将首先介绍如何搭建 Python 编程环境，然后在环境中创建第一个 Python 程序文件并运行。

1.3.1　搭建编程环境

打开 Python 官方主页“https://www.python.org/”，可以访问所有相关资源。

1．版本的选择

在主页上单击“downloads”，进入“https://www.python.org/downloads/”页面，可以看到当前最新版本，有 Python 2.x 和 Python 3.x 两个系列。Python 3.x 系列发布已达 8 年，目前已经非常成熟和稳定，全部的标准库和绝大多数第三方库都能够很好地支持 Python 3.x 系列，因此本书建议初次接触 Python 的读者，学习 Python 3.x 版本。本书内容也是围绕 Python 3 展开的。

2．下载并安装 Python 解释器

下载最新版本的 Python 解释器，单击图 1-4 中“Download Python 3.6.4”按钮，直接下载 Windows 操作系统的 Python 解释器，如果要下载其他操作系统的 Python 解释

器，单击相应的操作系统链接，如 Linux/UNIX、Mac OS X。

图 1-4　下载 Python 最新版本

双击下载的可执行文件“Python-3.6.4.exe”安装 Python 解释器，出现如图 1-5 所示的安装界面，勾选最下方的“Add Python 3.6 to PATH”复选框，选择“Install Now”将 Python 安装在默认路径下。如果要自行指定安装路径，选择“Customize installation”。等待安装过程，直至出现安装成功的界面。

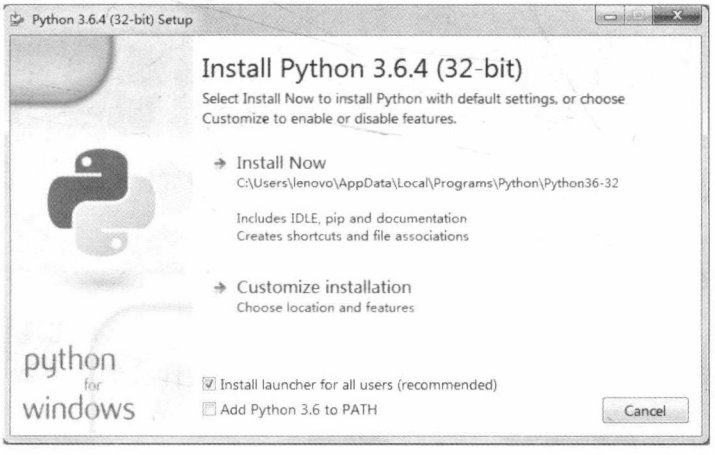


图 1-5　安装 Python 最新版本

安装完成后，在“程序”里就可以找到如图 1-6 所示的 Python 程序，其中 IDLE 是 Python 集成开发环境（Integrated Development Environment），也是最常使用的 Python 编程环境，Python 3.6 是 Python 命令行，也是常用的 Python 编程环境。

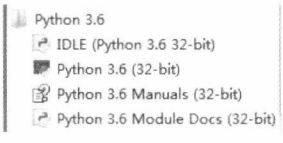


图 1-6　安装好的 Python 应用程序

3. 启动 Python 交互式解释器

通过 IDLE 方式启动 Python 交互式解释器如图 1-7 所示，通过命令行方式启动 Python 交互式解释器如图 1-8 所示。两种界面都以 3 个大于号“>>>”作为提示符，可以在提示符后输入要执行的语句，按〈Enter〉键执行。

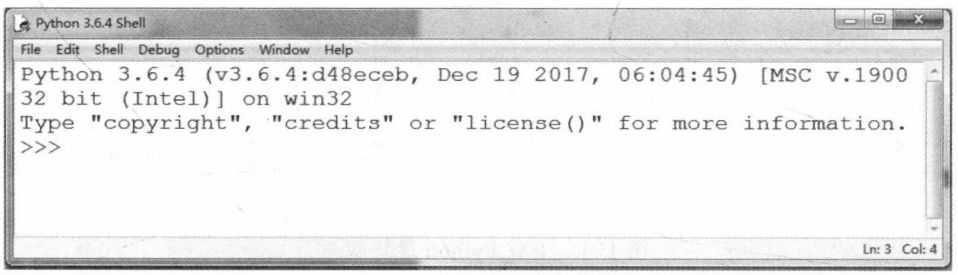


图 1-7　通过 IDLE 方式启动 Python 交互式解释器

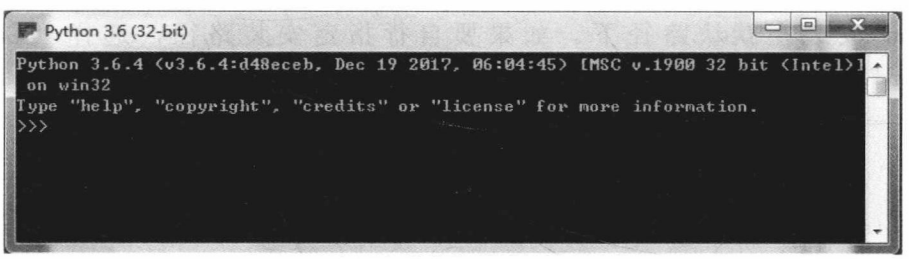


图 1-8　通过命令行方式启动 Python 交互式解释器

1.3.2　创建并运行程序

运行 Python 程序有两种方式：交互式和文件式。本书中凡是出现“>>>”，表示代码以交互式的方式运行，不带该提示符的代码则表示以文件式的方式运行。

1. 交互式

交互式指 Python 解释器即时响应用户输入的每条代码，给出输出结果。例如，在图 1-7 和图 1-8 的提示符后分别输入 print("Hello World")语句，按〈Enter〉键后，可以看到语句的运行结果是在屏幕上输出“Hello World”。交互式一般用于调试少量代码。在提示符“>>>”后输入 exit()或者 quit()可以退出 Python 运行环境。

2. 文件式

文件式是最常用的编程方式，也称为批量式，指用户将 Python 程序写在一个或多个文件中，然后启动 Python 解释器批量执行文件中的代码。在 IDLE 中，按快捷键〈Ctrl+N〉打开一个新窗口，或者在菜单中选择 File→New File 选项，在其中输入

Python 代码，并保存为.py 文件。例如，输入 print("Hello World")语句（图 1-9），在菜单中选择 File→Save As 选项，将文件保存为“hello.py”（图 1-10），选择想要保存文件的路径，单击“保存”按钮；按快捷键〈F5〉运行程序，或者在菜单中选择 Run→Run Module 选项，运行结果显示在 Python 交互界面中。

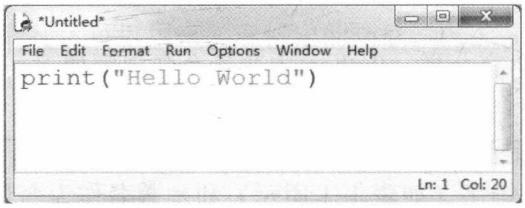


图 1-9　创建 Python 程序

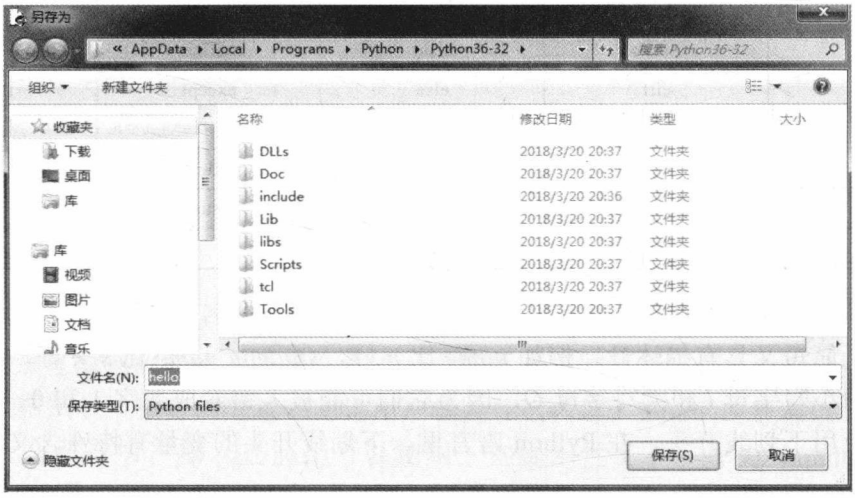


图 1-10　保存 Python 程序文件

1.4　Python 基础

本节将会介绍编写程序的一些基础概念，包括数据类型、变量、函数、语句等，理解了这些概念，才能开始编写程序。

1.4.1　数据类型

一种数据类型（type）是一系列值以及为这些值定义的一系列操作方法的集合。Python 的基本数据类型包括数字和字符串（详见第 2 章），如"Hello World"就是字符串数据类型。还有像列表、元组、字典这样的组合数据类型（详见第 4 章），如[1,2,3,4,5]就是一个列表。

1.4.2 变量

变量（variable）是编程语言中能存储计算结果或能表示值的抽象概念。在使用变量前，需要对其进行赋值，之后可以根据需要随时对变量重新赋值。

1. 变量的命名

与其他大多数高级语言一样，Python 中变量命名规则如下。

- 以字母或下划线（_）开头。
- 其他的字符可以是数字、字母或下划线。
- 不能将 Python 关键字（如表 1-1 所示）和函数名作为变量名。

表 1-1 Python 的 33 个关键字

False	None	True	and	as
assert	break	class	continue	def
del	elif	else	except	finally
for	from	global	If	import
in	is	lambda	nonlocal	not
or	pass	raise	return	try
while	with	yield		

除了以上必须遵守的命名规则以外，本书还给出如下命名建议。

- 应既简短又具有描述性，例如 *name* 比 *n* 好，*student_name* 比 *s_n* 好。
- 慎用小写字母 *l* 和大写字母 *O*，因为它们可能被人错看成数字 1 和 0。
- 避免用下划线开头，在 Python 语言里，下划线开头的变量有特殊含义。

2. 变量的赋值

变量赋值通过等号（=）来执行，如图 1-11 所示。1.2.2 节提到，Python 是一种动态语言，因此不需要预先声明变量的类型，变量的类型和值在赋值那一刻被初始化；同时变量的类型也是可以随时变化的，也就是说，可以先将某个变量赋值为一个字符串，然后又将其赋值为一个数字。

```
>>> myLastName = "Li"
>>> myFirstName = "Ming"
>>> myLastName
'Li'
>>> myFirstName
'Ming'
>>> print(myLastName)
Li
>>> print(myFirstName)
Ming
```

图 1-11 变量的赋值示例

在交互式解释器中，可以用变量名查看该变量的原始值。注意与 print()函数输出结果的不同，直接用变量名查看，字符串带引号，而用 print()函数输出时不带引号。

1.4.3　函数

函数就像小型程序一样，可以用来实现特定的功能。Python 有很多函数，用户也可以自己定义函数。本小节主要介绍输入输出函数，在后续章节中还会学习到很多函数。

1．print()函数

前面已经用到 print()函数，它可以用来输出字符信息，也可以以字符形式输出变量。print()函数的参数是要输出的内容，比如 print("Hello World")。如果要将字符常量和变量连接起来一起输出，可以使用字符串的拼接符“+”，如图 1-12 所示。如果想要一个字符信息重复出现多次，可以使用“*”，“*”前是想要重复的字符信息，“*”后是要重复出现的次数。

```
>>> print("Hello World! My last name is " + myLastName + ".")
Hello World! My last name is Li.
```

图 1-12　print()函数示例

2．input()函数

从用户那里得到数据输入的最简单的方法是用 input()函数，它读取标准输入，并将读取到的数据赋值给指定的变量，无论用户输入什么内容，input()函数都以字符串类型返回结果。在获得用户输入之前，可以使用一些提示性文字，作为 input()函数的参数，如图 1-13 所示。

```
>>> myLastName = input("Please input your last name: ")
Please input your last name: Li
>>> myLastName
'Li'
>>> help(input)
Help on built-in function input in module builtins:

input(prompt=None, /)
    Read a string from standard input.  The trailing newline is stripped.

    The prompt string, if given, is printed to standard output without a
    trailing newline before reading input.

    If the user hits EOF (*nix: Ctrl-D, Windows: Ctrl-Z+Return), raise EOFError.
    On *nix systems, readline is used if available.
```

图 1-13　input()函数示例

在交互式解释器中，如果不了解一个函数及其使用方法，可以调用 help()函数来获得帮助，如：help(input)。