高等学校计算机教材建设立项项目

Python
程序设计与实践
——用计算思维解决问题

李莹 主编　　焦福菊 孙青 编著

清华大学出版社

高等学校计算机教材建设立项项目

Python 程序设计与实践

——用计算思维解决问题

李莹 主编　焦福菊 孙青 编著

清华大学出版社
北京

内容简介

本教材主要讲授 Python 程序设计知识，采用案例教学和问题驱动的撰写方法，注重实践思维、计算思维和创新思维等教育理念与教材内容的结合。本教材将知识点和实际应用相结合，以教学案例引出理论讲解。案例源于现实生活，旨在让读者理解实际问题被抽象化、模型化和程序化的全过程。内容涵盖 Python 应用的各个方面，以对比方式阐述人和计算机在解决问题时的异同，让读者理解计算思维的本质。教材在设计上由易到难，分别阐述计算机如何描述和处理现实世界中的各类事物，如何表示各类事物之间的关系，如何组织和优化程序结构等，使读者能够将程序设计和现实问题相关联。在讲解某一知识点时，横向延伸与之相关的各类知识点；在讲解某一个案例时，纵向扩展该案例所能实现的各种功能模块，使读者能够比较全面、深入地理解问题和掌握知识。教材穿插了一些技巧性、实用性的说明，并且对重要代码添加了注释。本教材免费提供与内容相配套的教学课件和各个案例的程序源代码。

本教材的内容涵盖范围较广，案例贴近实际，既可作为以 Python 为基础的程序设计类课程的配套教材，又可作为学习 Python 的很好的自学参考书，也适合各层次 Python 开发人员阅读参考。

图书在版编目(CIP)数据

Python 程序设计与实践：用计算思维解决问题/李莹主编；焦福菊，孙青编著. —北京：清华大学出版社，2018
ISBN 978-7-302-47389-3

Ⅰ. ①P…　Ⅱ. ①李…　②焦…　③孙…　Ⅲ. ①软件工具—程序设计　Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字(2017)第 101920 号

责任编辑：张瑞庆
封面设计：何凤霞
责任校对：焦丽丽
责任印制：丛怀宇

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：三河市铭诚印务有限公司
经　　销：全国新华书店
开　　本：185mm×230mm　　**印　　张**：11　　**字　　数**：175 千字
版　　次：2018 年 6 月第 1 版　　**印　　次**：2018 年 6 月第 1 次印刷
印　　数：1～1500
定　　价：29.00 元

产品编号：069486-01

前　言

本教材主要讲授 Python 程序设计知识，包括良好的编程习惯、计算机描述现实事物、计算机处理现实事物、计算机的流程控制、计算机表示现实事物间关系以及程序编写方法共 6 章内容，涵盖了变量、数据类型、控制语句、数据结构、面向过程程序设计、面向对象程序设计、GUI 设计、网络编程和调试方法等重要知识点。本教材旨在让读者理解计算机解决问题的方法和思路，掌握程序设计的核心概念，构建基本的程序设计思想，学会编写中等难度的程序代码，为进一步学习和掌握计算机程序设计奠定良好的基础。

本教材采用“案例教学”和“问题驱动”的撰写方法，注重实践思维、计算思维和创新思维等教育理念与教材内容的结合。在编写时，按照问题求解的方式表述教学内容，不仅使学生掌握基本的程序设计知识，更重要的是教会学生解决问题的思维方法，即按照“提出问题、分析问题、讲解知识点、解决问题、总结思维方法”的思路组织教材内容。具体来讲，首先通过一个实例提出问题，然后分析解决问题的思路，引出解决该实例必须了解的核心概念和相关知识，并给出具体的解决方法，在完成具体的程序设计后，进一步展开阐述实用的编程技巧和工程实践经验，最后总结解决此类问题的思维方法，让学生不仅能够知其然，更能够知其所以然。撰写本教材的主要目标是学以致用，让学生掌握程序设计的基本技能，提高学生使用计算机解决实际问题的能力，同时更加注重学生计算思维和信息素养的培养，使他们具备用计算机抽象、分解、模拟和求解问题的能力，以及具备通过网络获取、分析和利用信息的自学能力。本教材的内容涵盖范围较广，案例贴近实际，既可作为以 Python 为基础的程序设计类课程的配套教材，又可作为学习 Python 的很好的自学参考书，也适合各层次 Python 开发人员阅读参考。

本教材的编写本着“案例引导知识、实践引导理论”的原则，将枯燥晦涩的理论性、原

理性的知识讲解转化成以问题驱动的案例教学。本教材设计和开发了一系列具有工程性、实践性、综合性等特点的教学案例。这些案例既联系所讲授的知识点，又注重学习者的学习兴趣，极大地激发了读者探究问题的求知欲。

本教材的主要特色和创新点如下：

(1) 案例丰富、贴近实际：将知识点和实际应用相结合，以教学案例引出理论讲解。案例源于现实生活，旨在让读者理解实际问题被抽象化、模型化和程序化的全过程。

(2) 内容全面、讲解独特：涵盖 Python 应用的各方面，以对比方式阐述人和计算机在解决问题时的异同，让读者理解计算思维的本质。

(3) 结构合理、设计新颖：教材以“用计算机解决现实问题”为主旨，在设计上由易到难，分别阐述计算机如何描述和处理现实世界中的各类事物，如何表示各类事物之间的关系、如何组织和优化程序结构等，使读者能够将程序设计和现实问题相关联。

(4) 难易适度、层层递进：教材采用横向和纵向两种方法撰写内容，在讲解某一知识点时，横向延伸与之相关的各类知识点；在讲解某一个案例时，纵向扩展该案例所能实现的各种功能模块，使读者能够比较全面、深入地理解问题和掌握知识。

(5) 代码注释、相关说明：为了便于读者的阅读和实现，教材穿插了一些技巧性、实用性的说明，并且对重要代码添加了注释。

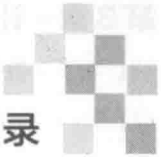
(6) 配套课件、案例源码：教材提供与内容相配套的教学课件和各个案例的程序源代码。

参与本教材编写的都是北京航空航天大学计算机学院从事计算机基础教学多年、有着丰富教学经验的老师。其中，第 1 章和第 4 章由焦福菊、李莹编写，第 2 章、第 3 章和第 5 章由李莹、孙青编写，第 6 章由孙青、李莹编写。全书由李莹统稿并定稿。此外，在本书的编写过程中得到了李宇川的极大帮助，并且参考了国内外许多同类的优秀教材，在此表示深深的谢意。

由于时间仓促，加之编者水平有限，所以尽管经过了多次反复修正，但书中仍难免会有疏漏和不足之处，恳请同行专家、一线教师及广大读者批评指正。

李　莹

2017 年 12 月于北京



目　录

第1章　良好的编程习惯

随着互联网的快速发展，计算机的应用已经遍布社会生活的各个领域，并逐渐改变着人们的生产和生活方式。现代社会，每个人都应该学会使用计算机。

今天，我们随处可见，快递小哥使用智能手机接外卖订单，餐厅服务员使用终端为顾客点餐，出租司机使用智能手机抢单拉活，学生使用数字图书馆浏览图书资料。然而，人们使用计算机做事的层次是不同的。快递小哥、餐厅服务员、出租司机、学生等大多数人都是普通用户，他们只需要知道有哪些应用程序可以为他们做事，学习如何使用这些程序即可；而专业技术人员则应该学习如何编写和优化能够解决实际问题的程序。

无论是进行卫星轨道、天气预报等复杂计算的超级计算机，便捷使用互联网的智能手机，还是控制冰箱、洗衣机的嵌入式计算机，对于使用者来说，它们都只是一个能够接收指令并输出计算结果的机器，如何进行计算的步骤则需要人们通过程序来告诉计算机。计算机程序就是人们告诉计算机如何完成预定任务的步骤。

计算机只认识0和1两个数字，我们如何告诉它怎样去完成任务呢？这就需要学习如何用计算机语言来编写程序。计算机语言种类繁多、各有特色，对于不同类型的应用程序，可能用某种语言编写程序更加方便或运行效率更高。例如，开发网站可以用PHP，控制网络数据传输可以用C语言，操作向量和矩阵可以用MATLAB，等等。每种语言都有基本的程序结构、语法和语义。例如，输入输出、基本的数学和逻辑运算、有条件地执行、重复执行，等等。学会了使用一门计算机语言编程，再学习其他的语言就会容易得多了。

Python 是一门功能强大、简单易学的通用高级编程语言，非常适用于计算机程序设计的教学和计算思维的训练。本书将带你学会使用 Python 语言来分析和解决实际问题，并在学习和实践中养成良好的编程习惯。

1.1 Python 简介

Python 语言由荷兰人 Guido van Rossum 于 20 世纪 80 年代发明，它是一种动态的解释型语言，具有面向对象特征。由于其语法简明易学、代码可读性高、程序清晰美观、可移植性强等优点，越来越受到编程者的欢迎。Python 语言具有如下一些特点。

(1) **自由软件**：Python 是免费而且开放源代码的程序设计语言，它遵循 GPL(GNU General Public License)协议，谁都可以自由地发布这个软件的拷贝，也可以阅读和改动它的源代码，并将它的一部分应用到其他自由软件中。

(2) **简单易学**：语言本身的组成成分较少，结构较小。提供交互式环境，对于学习编程的新手而言，Python 提供的实时反馈非常有帮助。

(3) **解释型语言**：Python 拥有自己的解释器，不用编译、链接等源代码到机器代码的转换过程。把 Python 程序复制到另外一台机器上，Python 可以直接从源代码执行程序。

(4) **程序可读性高**：Python 更接近于自然语言，易于阅读。例如，变量类型不用预先定义就可使用，它的代码的外观与内在语义紧密相关，有利于初学者一开始就养成良好的编程习惯，非常适合于教学。

(5) **面向对象**：Python 不仅支持面向过程编程，还支持面向对象编程。它是一种公共域的面向对象的动态语言。

(6) **可扩展性好**：在 Python 脚本中可以嵌入如 C/C++ 等其他语言编写的程序，也可以将 Python 脚本嵌入到 C/C++ 等其他语言编写的程序中。

(7) **可移植性好**：由于 Python 的开源本质，Python 已经被移植到如 Windows、Linux、FreeBSD、Macintosh、VxWorks、Windows CE 等很多操作系统平台上。如果程序

谨慎地使用依赖于系统的特性，则所有的 Python 程序都无须修改就可以在上述平台运行。

(8) **丰富的库**：Python 拥有丰富的标准库以支持各种功能应用程序的开发。例如，文档生成、线程、数据库、网页浏览器、电子邮件、文件传输、网络接口、图形界面等有关的操作。除了标准库以外还有很多库，例如，wxPython、Twisted 和图像库等。

由于上述特点，Python 越来越多地被用作初学者的入门编程语言。本书所有的代码都是在 Windows 64 位操作系统下安装的 Python 3.4.0 环境中运行通过的。

说明：Python 是一种动态、解释型语言。

(1) **动态语言和静态语言**：动态语言是指在程序运行时确定数据类型的语言。变量使用之前不需要类型声明，通常变量的数据类型是被赋值的数据的类型。静态语言是指在编译时由变量的数据类型即可确定的语言，多数静态类型语言要求在使用变量前必须显式地声明其数据类型。

对于动态语言，变量可以在程序的不同位置被赋予具有不同数据类型的数据(数值型、字符串型等)；而对于静态语言，一旦变量被指定了某个数据类型，如果不经过强制类型转换，它将永远保持这个数据类型。

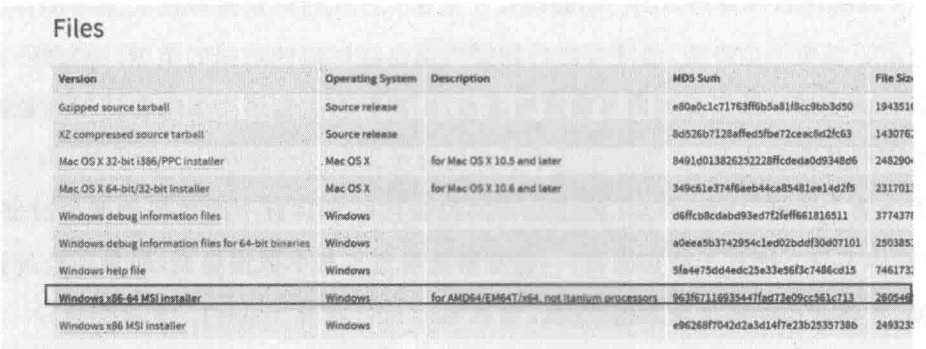
(2) **解释型语言和编译型语言**：解释型语言是在程序运行时，由与语言配套的解释器将程序逐条翻译成机器语言，即边解释边执行。解释型语言每执行一次就要翻译一次，效率比较低。编译型语言是在程序运行前，由与操作系统配套的编译器将程序整体翻译成机器语言，即一次编译、任意执行。编译型语言只需要在运行前完成一次翻译(原程序有变动除外)，而在运行时不需要翻译，程序的执行效率较高。

编译型语言与解释型语言，两者各有利弊。编译型语言由于程序执行速度快，同等条件下对系统要求较低，因此像开发操作系统、大型应用程序、数据库系统时都采用编译型语言，像 C/C++ 等语言基本都可视为编译型语言；而一些像网页脚本、服务器脚本及辅助开发接口这样的对速度要求不高、对不同系统平台间的兼容性有一定要求的程序，则通常使用解释型语言，如 Java、JavaScript、Python 等都是解释型语言。

1.2 Python 安装

1. Python 的安装

Python 语言的官方网站 www. python. org 为人们提供了免费的安装程序和学习文档。由于 Python 可以运行在多种操作系统平台，在下载 Python 时要注意自己的计算机所使用的操作系统和处理器的位数。例如，计算机是 64 位、运行 Windows 系统，则下载的文件应该是如图 1-1 方框中的 Windows x86-64 MSI installer。



Files

Version	Operating System	Description	MD5 Sum	File Size
Gzipped source tarball	Source release		e80a0c1c71763ff6b5a81f8cc9bb3d50	19435
XZ compressed source tarball	Source release		8d526b7128affed5fbe72ceac8d2fc63	14307
Mac OS X 32-bit i386/PPC installer	Mac OS X	for Mac OS X 10.5 and later	8491d013826252228ffcdeda0d9348d6	24829
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	349c61e374f6aeb44ca85481ee14d2f5	23170
Windows debug information files	Windows		d6ffcb8cdabd93ed7f2feff661816511	37743
Windows debug information files for 64-bit binaries	Windows		a0eea5b3742954c1ed02bddf30d07101	25038
Windows help file	Windows		5fa4e75dd4edc25e33e56f3c7486cd15	74617
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64, not Itanium processors	963f671169354477fad73e09cc561c713	26054
Windows x86 MSI installer	Windows		e96268f7042d2a3d14f7e23b2335738b	24932

图 1-1　Python 文件下载

Windows 64 位操作系统下安装的过程很简单，按照提示安装即可。安装完成后，生成两种 Python 运行环境，一种是命令行方式，另一种是 IDLE 方式。IDLE 是一个综合编辑、运行、调试的集成开发环境，如图 1-2 所示，本书后面的实例都使用 IDLE 方式。

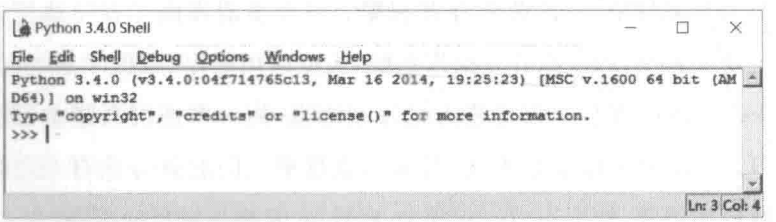


图 1-2　IDLE 集成开发环境

注意：尽量将 Python 安装在全英文路径下。

2. Python 初体验

现在从最简单的输出字符串“hello world!”入手来体验 Python 语言的魅力。

Python 的程序代码又称为脚本，是一系列指令的集合。Python 是动态、解释型语言，因此它拥有自己的解释器，每一条 Python 指令都可以直接执行。在 IDLE 集成开发环境中(Shell 窗口)输入一条指令，按回车键后，它会立刻显示结果。图 1-3 是 Python 3.4.0 命令行窗口，“>>>”是 Python Shell 的提示符。在提示符后面输入如下的指令：

```
>>>print("hello world!")
```

Python 立即将要打印的字符输出在屏幕上。print()是一个内置函数，输出指定的变量、函数或字符串的值。

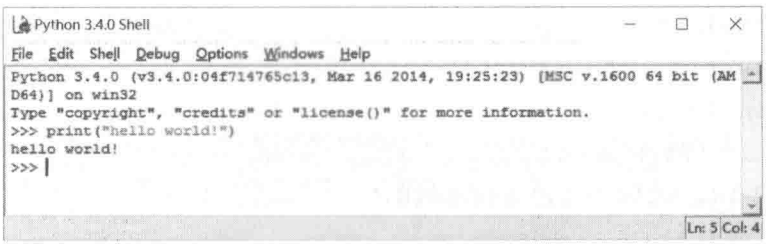


图 1-3　Python 3.4.0 命令行窗口

有时可以像使用计算器一样执行一些简单的运算。例如，图 1-4 所示的语句，第一条语句将数值 10 赋给变量 a，第二条语句将数值 100 赋给变量 b，第三条语句计算 a 加 b 的

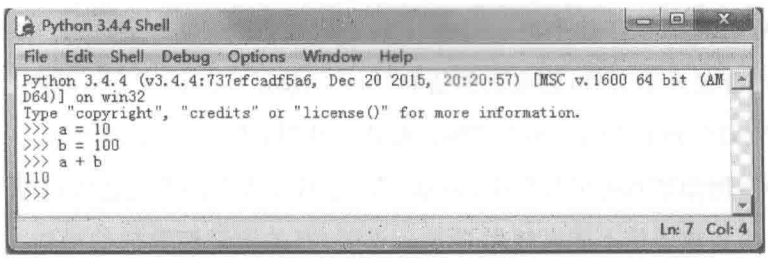


图 1-4　简单的 Python 语句

和。执行这三条指令后，Python 输出 a+b 的结果。

输入下面的指令会输出什么结果呢？请读者自行练习。

```
>>>21 * 30
>>>5+100.30
>>>1+9j+3
>>>2+8j * 3
>>>(5+3j) * 3
```

从上面几个实例可以看出，Python 不仅支持整数运算、浮点数运算，还可以支持复数运算。加、减、乘、除等运算符号是有优先级的，括号可以改变运算符的优先级。有关数据类型的各种操作将在第 2 章中讨论。

再输入下面的指令，看看又会输出什么结果。

```
>>>s="hello"
>>>t="world  ! "
>>>print(s+t)
>>>a=    'Is it a cat? '
>>>b=" 'No,it's not.' she said."
>>>print(a+b)
>>>c=100
>>>d="123"
>>>print(c+d)
```

这些实例说明 Python 不仅可以处理数字，也可以处理字符串。在 Python 中，字符串是由双引号或单引号括起来的字符组成的，这里的字符包括字母、数字和控制字符。如果字符串本身包含单引号，则上面的语句“b=" 'No,it's not.' she said." ”是正确的，而写成“b='"No,it's not." she said.'”则会报错。这时，我们也可以用转义字符“\”来表示中间的单引号，例如“'"yes,it\'s " she said.'”。字符串类型的数据也可以做连接、截取等操作，但字符串和数值是不能直接做算术运算的，有关字符型数据的操作将在第 2 章中讨论。

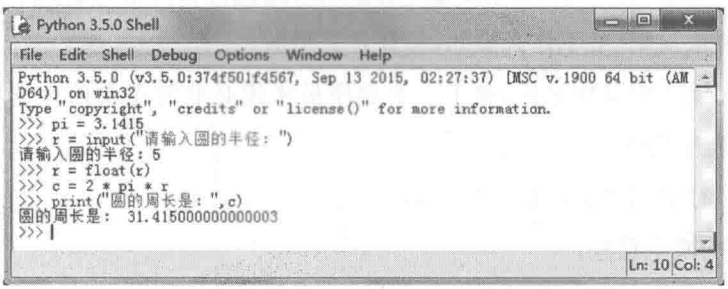
3. 编写 Python 程序

Python 提供两种编写程序的方式：

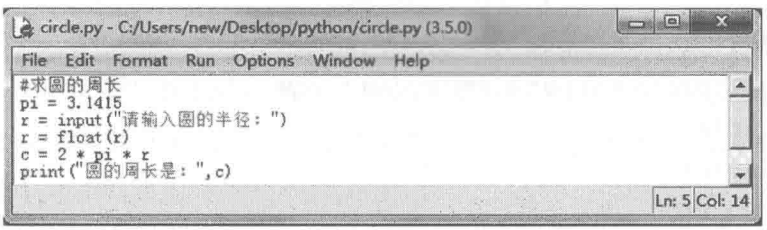
(1) Shell 窗口编写；

(2) 文件窗口编写。

Shell 窗口提供一种交互式的编程模式，它一般用于编写简单的程序，如图 1-5(a)所示。Shell 窗口的特点是“边输入指令、边执行并输出结果”，即输入的每条 Python 指令会在按下 Enter 键后被立即执行。只要不打开新的 Shell 窗口(IDLE)，前面定义的变量在后面的指令中都可以使用。一旦关闭 Shell 窗口，会话中的所有变量和输入的语句就不存在了。为了使程序代码能够被重复执行，需要将代码保存到文件中。和其他编程语言一样，我们需要用文件窗口来编写和保存源代码，如图 1-5(b)所示。文件窗口的特点是“输入完整代码后一次执行并输出结果”。Python 源代码文件是普通的文本文件(*.py)，可以用任意能够编辑文本文件的编辑器来编写 Python 程序，如记事本、Word 等。



(a) Shell窗口



(b) 文件窗口

图 1-5　Python 集成开发环境

在 IDLE 中单击 File|New File 菜单，在打开的可编辑环境中输入 Python 语句，并保存为 *.py 格式文件，这样就可以实现程序的反复查看、修改和重复执行了。

另外，Python 的 IDLE 还有内容的高亮显示功能，即可以根据输入内容（如程序注释、关键字等）的不同而自动识别并显示不同的颜色，使得程序显得更加清晰、易读。本书后面的实例如果没有特别说明，则是在文件中编辑、编译和执行的。

1.3 漂亮的程序

程序是一件艺术品，一个符合规范的程序是“十分漂亮的”。这里，“漂亮”有以下两层含义：

（1）满足编程语言的语法规则：在 Python 中，体现代码层次关系的缩进（4 个空格）和冒号“:”都是语法规则，不能省略。

（2）符合阅读程序的审美习惯：编程时，为了提高程序的可读性和可维护性，通常会对关键语句添加注释（以 # 开头的语句），也会在不同代码块间增加空行。这些操作不属于 Python 的语法规则，虽不是必需的，但却是常用的。

可见，养成规范的编程习惯，对于一个程序员来说是非常重要的。下面试着读一读例 1-1 给出的程序代码。

【例 1-1】 计算两个数的最大公约数。

对比下面的两个程序。

（1）无层次区分的程序

```
a=input("please input the first number:")
b=input("please input the second number:")
num1=int(a)
num2=int(b)
if num1<num2:
temp=num1
num1=num2
```

```
num2=temp
while num2 !=0:
temp=num1 %num2
num1=num2
num2=temp
print(a,"和",b,"的最大公约数是:",num1)
```

(2) 有层次区分的程序

```
a=input("please input the first number:")
b=input("please input the second number:")
num1=int(a)
num2=int(b)

if num1 <num2:
    temp=num1
    num1=num2
    num2=temp

while num2 !=0:
    temp=num1 %num2
    num1=num2
    num2=temp

print(a,"和",b,"的最大公约数是:",num1)
```

先不用急于读懂代码,仅从形式上对比上面的两个程序,就能直观地感受到有层次区分的程序在代码可读性、结构逻辑性等方面的优势。另外,无层次区分的程序不但难以理解,更重要的是其执行时会报错。究其原因,是因为它不符合 Python 的编写规范(详见 1.3.1 节)。

说明:(1) 空格的使用:空格常被用来修饰程序,使其看起来更加漂亮且易于阅读。一般会在二元操作符两边。例如,赋值操作符(=),比较操作符(==、<、>、!=、<>、