普通高等教育“十三五”规划教材
新工科建设之路·计算机类专业规划教材

# Python 3

## 从入门到实战

董洪伟 编著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

# Python 3
## 从入门到实战

- **语言简单易懂、精练清晰**

用简单易懂、精练清晰的语言描述语法概念和相关知识，避免冗长、晦涩。

- **内容全面、案例典型**

提供了数据结构、信息管理、游戏编程、机器学习、强化学习等不同领域的典型案例，并介绍了常用标准库。

- **资源丰富、高效学习**

除提供课程源代码和教学PPT外，还提供配套教学视频、教学网站等，帮助读者高效学习。相关资源可登录华信教育资源网（www.hxedu.com.cn）下载。

策划编辑：戴晨辰
责任编辑：张　慧
封面设计：田晨晨

ISBN 978-7-121-35356-7

定价：75.00元

普通高等教育“十三五”规划教材
新工科建设之路·计算机类专业规划教材

# Python3

## 从入门到实战

● 董洪伟 编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

## 内 容 简 介

本书是一本语法与实践相结合的 Python 入门教程，全书分为上、下篇。上篇为“Python 语法与实践”，以简明的语言、易懂的案例介绍 Python 的变量与对象、运算符与表达式、控制语句、函数、内置数据类型，Python 的面向对象特征，如类与对象、派生类、类的实例与静态方法等 Python 语言的核心语法，以及迭代器与可迭代对象、闭包、装饰器、@property、深拷贝与浅拷贝等高级语言特征，还介绍了错误与异常、调试。在核心语法部分采用来自数据结构、游戏编程、信息管理、机器学习、强化学习等其他学科和领域的一些经典问题作为实战演练，展示了 Python 解决实际问题的强大功能，以提高初学者的实际编程能力，使其尽快熟悉语法的使用。下篇为“Python 标准库”，对常用的一些 Python 标准库，如操作系统接口模块、时间日期模块，以及正则表达式、并发计算、图形用户接口编程、网络套接字编程、Internet 应用编程、数据持久化等进行了介绍。

本书描述精练、通俗易懂，提供了丰富的实战案例，既可作为大学本科和高职高专相关专业课程的教材，也可供编程爱好者学习和参考。



**图书在版编目（CIP）数据**

Python3 从入门到实战 / 董洪伟编著. —北京：电子工业出版社，2020.1
ISBN 978-7-121-35356-7

I. ①P… II. ①董… 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(2018)第 251247 号

策划编辑：戴晨辰
责任编辑：张 慧
印 刷：北京虎彩文化传播有限公司
装 订：北京虎彩文化传播有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编：100036
开 本：787×1092 1/16 印张：26.75 字数：684.80 千字
版 次：2020 年 1 月第 1 版
印 次：2020 年 1 月第 1 次印刷
定 价：75.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：dcc@phei.com.cn。

# 前言

Preface

Python 是一种易于学习、功能强大的编程语言。它具有高效的高级数据结构，能够简单有效地实现面向对象编程。Python 简单的语法和动态类型，连同解释型特性，使其成为不同平台上脚本处理及快速应用开发的理想语言。此外，Python 还是数据分析和人工智能的首选编程语言。

编著者希望编写一本既简明扼要，又深入全面的 Python 教材，既避免过多的语法细节，又注重语言本身实际使用能力的培养。

本书突出重点，讲解主要的常用语法，而不是面面俱到的语法细节。全书由浅入深，由易到难，尽量用浅显易懂的例子说明语法概念，力求简明扼要，避免空洞的概念和冗长的描述，帮助无编程基础的初学者在较短的时间里快速理解 Python 语言的核心特征。

只有通过具体、长期的实战训练，才能逐步精通一种编程语言。语法知识可以在短期内学习并理解，但只有经过大量实战训练才能真正熟练掌握一种编程语言。本书准备了游戏编程、信息管理、数据结构、机器学习、强化学习等不同领域的经典实战案例，希望可以通过这些案例，帮助读者消化语法知识、提高学习兴趣，逐步将 Python 用于解决各种实际问题而不是用于简单的语法练习，希望避免“只会考试而不会编程”的普遍问题。

实战案例涉及一些其他学科的专业知识，初学者或教师可以根据自己的需要选读或选讲实战部分。

本书包含配套学习资源，读者可在本书的 github 网站(https://hwdong-net.github.io)或登录华信教育资源网(www.hxedu.com.cn)注册后免费下载。

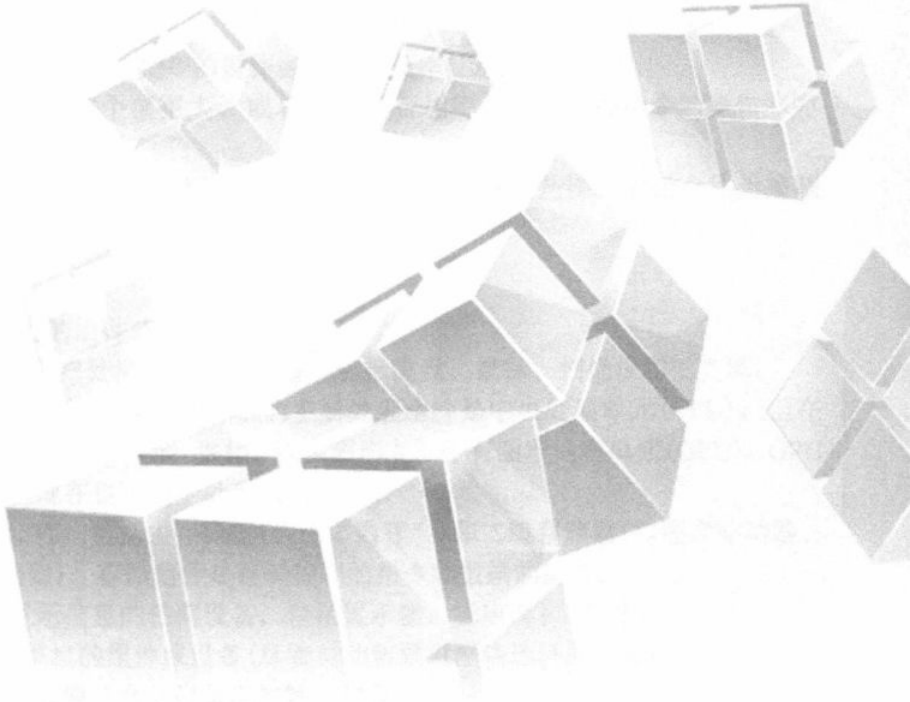
由于编著者水平所限，书中错误之处在所难免，欢迎读者对本书进行批评与指正，共同完善本书内容，使更多的读者受益。

编著者

# 目录

## 上篇 Python 语法与实践

# Python 语法与实践



# 第1章 Python介绍

## 1.1 程序与编程语言

### 1.1.1 计算机是什么

计算机是一种根据指令对数据进行处理的通用计算设备。每台计算机都有一个称为“中央处理单元(CPU)”的微处理器芯片用于执行对数据进行处理的指令，不同计算机的指令集是不一样的。

#### 1. 计算机指令

计算机接收一系列指令作为输入，然后逐个处理它们，最后输出某些信息以显示它已完成的操作。这一过程类似人们日常生活中通过一系列操作步骤完成一个任务的过程。例如，一个人通过下列步骤完成“做饭”的任务：

```
从容器(米桶)中取出米，放入洗米盆；
用水对洗米盆中的米进行冲洗；
如果电饭煲没洗净
    清空并洗净电饭煲；
打开电饭煲的盖子，将米和水放入电饭煲；
插上电源，按下开关；
饭做好后，拔下电源(任务结束)。
```

虽然人们可以理解自然语言(如英语)中的复杂指令，但计算机只能理解用计算机语言表达的非常简单的机器指令集中的指令。无论多么复杂的计算，在计算机内都会被分解成许多条简单的可逐条执行的机器指令。告诉计算机如何执行复杂任务的指令序列称为**程序**。

以下是一些简单的计算机指令示例。

- 算术：加、减、乘或除。执行这些指令的操作称为**算术操作**(运算)。
- 比较：比较两个数字，查看哪个值更大，或者它们是否相等。执行这些指令的操作称为**逻辑操作**(运算)。
- 分支：跳转到程序的其他指令处，并从那里继续运行程序。执行这些操作的指令称为**控制语句**。

#### 2. 计算机的组成部分

计算机包含以下四种主要类型的组件或设备。

- 输入设备：允许计算机从用户处接收信息的设备，包括键盘、鼠标、扫描仪和麦克风。
- 处理组件：处理信息的计算机组件。计算机的主要处理组件是中央处理单元(CPU)，但在现代计算机中也可能有其他处理单元。例如，许多图形卡都带有图形处理单元(GPU)，GPU以前只用于处理图形，但现在也可用于处理通用程序。
- 存储组件：存储信息的组件，包括主存储器(也称“**内存**”)和二级存储器(如硬盘驱动器、CD或闪存盘等外部存储器)。存储组件是存储程序的指令和数据的地方。
- 输出设备：用于向用户显示信息的任何设备，包括显示器、扬声器和打印机。

可以用自动售票机来理解计算机的组件或设备(尽管自动售票机严格地讲并不是计算机)。

- 输入设备：投币口和选择按钮是自动售票机的输入设备。

- 处理组件：当使用者进行选择时，自动售票机执行的操作包括验证是否有满足条件的票；验证身份信息；检查和验证是否收到足够的资金；修改数据库；计算差额。执行所有这些操作的机器部分可以看作处理组件。
- 输出设备：显示结果；打印票据；退回多余资金。
- 存储组件：保存销售数据及价格等信息。

#### 3. 中央处理单元(CPU)

CPU 是计算机中最重要的部分，是计算机的“大脑”，主要负责计算、处理数据、控制其他设备等工作。它有以下几个重要的子组件。

- 算术/逻辑单元(ALU)：执行算术和比较运算。
- 控制单元：确定下一个要执行的指令。
- 寄存器：形成一个高速存储区以保存临时的运行结果。

不同种类的 CPU 可以处理不同的指令集，如 Intel IA-32、x86-64、IBM PowerPC 或 ARM 等。

#### 4. 存储器(Memory)

计算机将信息(程序、数据)存储在存储器中，存储器分为两类，**主存储器**(也称**内存**)和**辅助存储器**(也称**外存**)。



**主存储器**直接连接 CPU(或其他处理单元)，通常称为**随机存取存储器(RAM)**。计算机关闭时，大多数主存储器都会丢失其内容，即具有“易失性”。

可以将主存储器看作一组一排乘一列的存储器单元，每个存储器单元都可以通过其存储器地址寻址。第一个单元的地址为零，并且每个后续单元的地址比它之前的地址多一个，正如一个班级的学生的学号从 1 开始依次递增一样。每个存储器单元只能保存长度固定的用二进制表示的数值，但 CPU 可以随时用新的数值替换原有数值。

**辅助存储器**比主存储器价格便宜，但可以存储更多内容。虽然辅助存储器的存储速度比主存储器慢得多，但它是非易失性的，也就是说，即使在计算机关闭后其保存的信息也会保留，如硬盘和闪存盘。

计算机的操作系统提供操作辅助存储器的高级接口，这些接口允许信息以文件的形式保存在辅助存储器中，并且文件组织呈目录式的层级结构。接口和层级结构通常称为“文件系统”。

### 1.1.2 计算机编程

#### 1. 算法(Algorithms)

算法是完成某个任务或解决某个问题的一系列步骤(指令)。

#### 2. 编程和程序

编程就是使用计算机的指令来表示算法，即将算法转换为计算机可以执行的程序。程序就是算法在计算机中的表示和实现。

#### 3. 二进制

因为计算机硬件是由很多晶体管元器件组成的，而晶体管元器件只有“开”和“关”两种状态，所以 1 个晶体管只能表示两个数字：0 和 1。通过多个表示 0 或 1 的晶体管元器件可以组合出更复杂的数值，如整数或字符等。无论多么复杂的程序数据，在计算机硬件中都是以二进制(0 和 1)的形式表示的。

1 个晶体管元器件只能表示 1 位二进制数(0 或 1)，称为 1 **比特**(**Bit**)，简记为 b 或 1 位。8 个晶体管元器件可以表示 8 位二进制数字，即可表示 $2^8$ 个不同的数值。8 位二进制数，称为 1 **字节**(**Byte**，简记为 B)。16 个晶体管元器件可以表示 16 位二进制数，即 2B。

```
8×1024 个晶体管元器件可以表示 1024B，即 1KB。
8×1024×1024 个晶体管元器件可以表示 1024KB，即 1MB。
8×1024×1024×1024 个晶体管元器件可以表示 1024MB，即 1GB。
```

#### 4. 机器语言(machine language)

计算机中的指令和数据都是用 0 和 1 表示的。机器语言是用这种二进制数表示的、计算机能够直接识别和执行的机器指令集合。

例如，下列是将 17 和 20 相加的机器指令(采用 Intel 8086 机器语言，Intel Pentium 机器语言的子集)：

```
1011 0000 0001 0001
0000 0100 0001 0100
1010 0010 0100 1000 0000 0000
```

第一行告诉计算机将 17 复制到 AL 寄存器：前 4 位二进制数(1011)告诉计算机将信息复制到寄存器中，接下来的 4 位二进制数(0000)告诉计算机使用名为 AL 的寄存器，最后 8 位二进制数(0001 0001，表示整数 17)为要复制的数。

用机器语言编写程序非常困难，也很难被人们阅读和理解，但在 20 世纪 40 年代，第一台计算机的程序员必须这样做，他们通过纸上打孔表示 0 和 1 来编写机器语言的程序，因为没有其他选择！

#### 5. 汇编语言(assembly language)

为了简化编程过程，人们引入了汇编语言。每条汇编指令对应一条机器语言指令，使人们更容易理解，如上述的机器语言用 8086 汇编语言中等效的加法程序可写为：

```
MOV AL, 17D
ADD AL, 20D
MOV [SUM], AL
```

用汇编语言编写的程序不能直接被计算机理解，因此需要翻译步骤。一种叫作“**汇编程序**”的程序可以将汇编语言编写的程序转化为机器语言程序。

#### 6. 高级语言(High-level language)

虽然汇编语言对机器语言有很大的改进，但它仍然很神秘，而且它的级别太低，和机器语言一样，即便是完成最简单的任务也需要很多条指令。于是，人们发明了高级语言，使编程变得更加容易。

在高级语言中，一条指令可以对应多条机器语言指令，高级语言采用类人类的语言表示指令，这使得程序更易于读取和写入。以下是上述代码的 Python 等效代码：

```
sum = 17 + 20
```

### 1.1.3 编译器、解释器和 Python 语言

用高级语言编写的程序，在计算机执行之前也必须翻译为机器语言。某些编程语言的程序首先被整体翻译为机器语言的程序，并存储在指定文件中，然后再执行，这种语言称为**编译型语言**。也某些语言的程序语句被逐行翻译后再逐行执行，这种语言被称为**解释型语言**。Python 就是解释型语言。

编译型语言通过一个**编译器**程序，将编译型语言编写的源文件编译为可执行的二进制程序文件。解释型语言通过一个**解释器**程序，对解释型语言编写的源文件的每一条语句逐条解释并执行。

- **编译器**：编译器是将整个源代码程序一次性全部转换为机器语言代码的工具。转化后的机器语言代码可以直接在计算机上运行。
- **解释器**：是“一行一行”地解释执行，即将每条语句翻译成解释器自己的中间代码，然后再由解释器执行这个中间代码。每条语句是在解释器自己的虚拟环境中执行的。

解释器对每条语句逐条转换执行，使初学者很容易知道程序的错误位置。而编译器对整个源程

序进行一次性的转换，其优点是可以对代码进行整体的优化，从而提高程序的性能。

要求运行速度高或需要直接操纵硬件的程序通常使用 C 或 C++语言编写。C 或 C++语言是编译型语言，编译器将 C 或 C++语言程序编译为机器指令时，会对程序做很多细粒度的控制和优化，从而可以提高程序的运行速度，但编译需要耗费时间并且容易出错，而且需要整个程序写完且编译好后才能开始执行程序，而 Python 作为一种解释型语言，每条语句被逐条解释执行，可以立即发现程序错误，并可以看到程序运行结果，因此学习和编写这种解释型语言更容易。

### 1.1.4 Python 程序开发步骤

利用 Python 编写程序要经历以下几个步骤。

(1)理解问题：如这是一个什么样的问题？输入数据是什么？要产生什么结果？

(2)提出算法：解决这个问题的指令(步骤)序列。

(3)编写程序：将算法转换成某种编程语言的程序。

(4)测试：输入各种可能的不同的数据，检测是否产生预期的结果。

例如，要计算一组数值的平均值，可以按照上述步骤进行。

(1)理解问题：这些数值从哪里输入(键盘还是文件)？结果如何显示(屏幕打印输出还是保存到文件)？



(2)提出算法：首先用两个数值分别表示总和(sum)及数值的个数(计数器，count)，然后将输入的数值累加到总和上，同时累计数值的个数，最后用总和除数值的个数，得到平均值。人们经常以一种“伪代码”的方式描述算法的过程：

```
-----------start-------------
总和 sum=0。
计数器为 count=0。
重复:
    读一个值,
    如果读取值失败，结束这个“重复”过程,
    否则:
        将读取的值 value 加到 sum。
        计数器增加 1。

通过“总和”及“计算器”相除得到平均值。
显示/打印平均值。
-----------end------------
```

(3)编写程序：将算法用 Python 语言表示出来。

(4)测试： 输入不同的测试数据，查看结果是否正确。输入的数据可以包含非法数据，查看程序能否适当地应对。例如，输入的数据是字符串而不是数值，程序是否会提示等。

## 1.2 Python 语言的主要特征

Python 语言是由 Guido Van Rossum 于 1991 年发明的高级通用编程语言。所谓高级，是指它隐藏了机器指令的底层细节，并且提供给程序员的是一种人类易于理解的编程语言的语法。所谓通用，是指它可以开发各类应用程序。和其他高级编程语言，如 C、C++、Java 等不同，Python 语言以其简单易懂和编程效率高而著名。

Python 语言的主要特征如下。

**1. 简单易学**

Python 语言简单，特别适合初学者学习和使用，使初学者可以把精力集中在问题本身和求解方

法上，而不用担心语法、类型等外在因素。Python 语言对代码的逐句解释执行方式，可以帮助初学者非常容易发现程序错误。正是由于其简单性，国内外很多教育机构都将 Python 语言作为中小学的首选编程语言。

**2. 编程效率高**

同一个任务，用 Python 语言编写的代码往往是用其他编程语言所编写的代码的代码量的几分之一甚至几十分之一。例如，有时，用其他编程语言可能需要编写几百行甚至几千行代码，用 Python 可能只需编写几行代码。

**3. 跨平台**

Python 语言是一个跨平台的编程语言，用该语言编写的程序可以在各种操作系统，如 Windows、UNIX/Linux、Mac，甚至手机操作系统，如 Android 等，平台上运行。

**4. 解释性语言**

Python 语言是一个解释型编程语言，而 C、C++则是编译型编程语言。

**5. 大量的库**

Python 语言提供大量丰富的库，包括 Python 语言自带的标准库和第三方的库。利用这些大量优秀的库，可以避免程序员重复“造轮子”。



由于 Python 语言的这些优点，Python 语言不再是程序员的专利，各行各业的人都在使用 Python 语言处理他们的数据和业务。特别是处理大数据和人工智能，都普遍采用 Python 语言作为其编程开发的语言。目前，世界范围内掀起了学习 Python 语言的热潮，从计算机专业到其他专业的研究开发人员，从大学生到中小学生，都在学习 Python 语言，国内许多大学、中学、小学都已经将 Python 语言作为首选编程语言。教育部考试中心于 2017 年 10 月 11 日发布了《关于全国计算机等级(NCRE)体系调整》的通知，决定自 2018 年 3 月起，在计算机二级考试加入“Python 语言程序设计”科目。

在 2017 年 IEEE 编程语言排行榜中，Python 语言排名第一。

## 1.3 Python 开发环境及安装方式

### 1.3.1 安装 Python

Python 开发环境的安装方式通常有两种方式，原生安装和工具包安装。

**1. 原生安装**

原生安装是指只安装相应平台最基本的 Python 解释器。在 https:// www.python.org/downloads 下载 Python 安装程序并运行即可。

如图 1-1 所示，在安装过程中勾选“Add Python 3.6 to PATH”，安装程序会自动将 Python 解释器的路径添加到系统路径。

安装完成后，可以打开终端窗口，在其中输入“python”就可以打开 Python 解释器，系统将显示 Python 的版本信息：

```
>>> python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit
        (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
```

原生安装只安装最基本的 Python 解释器和一些自带的 Python 库。