



高等院校计算机类规划教材



Python

人工智能基础

主 编 孙成立 赵敏 李持
副主编 高建波 方芳 廖远 涂继亮



北京邮电大学出版社
www.buptpress.com



高等院校计算机类规划教材

Python 人工智能基础

主 编 孙成立 赵 敏 李 持
副主编 高建波 方 芳 廖 远 涂继亮



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本教材主要涉及人工智能的核心内容——机器学习,主要内容如下所述。

绪论:介绍人工智能的概念、特征、发展历史及人工智能、机器学习、深度学习的关系。

Python 基础:介绍开发环境,包括元组、列表、字典、集合,模块和函数,科学计算库 NumPy,数据分析工具 Pandas,数据可视化,面向对象编程、Python 的 GUI 程序设计及 Python 的数据库编程。

有监督学习算法:K 近邻算法、逻辑回归、决策树、支持向量机等。

无监督学习算法:K 均值算法、基于密度的扫描聚类、高斯聚类模型、主成分分析。

集成学习算法:非强依赖的 Bagging 和 随机森林算法及强依赖的 Boosting 算法(着重讲解 XGBoost 算法)。

神经网络和深度学习:神经网络的基本概念和深度学习基础。

本教材主要面向对 Python 人工智能基础感兴趣的初学者。掌握本教材内容以后,读者可以深入学习深度学习的相关内容。

图书在版编目(CIP)数据

Python 人工智能基础 / 孙成立, 赵敏, 李持主编. -- 北京: 北京邮电大学出版社, 2022. 6

ISBN 978-7-5635-6654-9

I. ①P… II. ①孙… ②赵… ③李… 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(2022)第 085883 号

策划编辑: 刘纳新 姚 顺 责任编辑: 王晓丹 米文秋 责任校对: 张会良 封面设计: 七星博纳

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号

邮政编码: 100876

发行部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 唐山玺诚印务有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 19.5

字 数: 483 千字

版 次: 2022 年 6 月第 1 版

印 次: 2022 年 6 月第 1 次印刷

ISBN 978-7-5635-6654-9

定价: 46.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

随着人工智能理论研究和实践应用的不断深入发展,人工智能基础知识愈显重要。与十多年前甚至更早时期相比,人工智能基础的相关内容已有所变化,迫切需要更新相关教材。

在内容编排上,本教材以机器学习为主要讲解内容,对神经网络和深度学习只做了初步介绍。

与传统教材侧重高等数学、统计学等学科的公式推导相比,本教材从最简单的案例入手,力求将枯燥的理论用通俗易懂的语言深入浅出地进行讲解,避免了深奥又枯燥的公式推导拉高初学者的进入门槛,极大地增加了初学者的学习信心。

不同于有些教材专注于编程用法而回避人工智能理论基础的特点,本教材在重点编排理论知识的同时,运用大量案例,从简单到复杂,阶梯式逐步引导,确保初学者在增强学习人工智能的乐趣的同时,领会人工智能相关前沿知识。

此外,本教材所用数据源多数是从 `sklearn.datasets` 中导入的数据集,极大地解决了初学者难以找到数据源的苦恼。

相信通过对本教材的学习,越来越多的年轻学子将掌握人工智能技术的初步知识,为进一步研究人工智能打下坚实基础,并逐渐运用人工智能技术,发展人工智能技术,为我们的民族、我们的国家贡献一份力量。

本教材由孙成立负责规划全书布局、章节设定等工作;赵敏负责完成多个章节的理论部分的撰写;李持负责编写神经网络和深度学习等的相关内容;高建波、方芳、廖远、涂继亮负责完成各个章节的资料查找及代码调试工作。

由于编者水平有限,书中难免存在疏漏和不足之处,恳请读者批评指正。

目 录

绪论	1
第 1 章 Python 基础	5
1.1 开发环境、Python 基础知识	5
1.1.1 开发环境	5
1.1.2 Python 基础知识	5
1.2 列表、元组、字典、集合	10
1.3 模块和函数	18
1.4 科学计算库 NumPy	25
1.5 数据分析工具 Pandas	39
1.6 数据可视化	41
1.7 面向对象编程、Python 的 GUI 程序设计及 Python 的数据库编程	45
第 2 章 模型评估与选择	70
2.1 模型的评估方法	70
2.2 性能度量的方法	71
2.3 模型的选择与常见模型的优缺点	73
第 3 章 K 近邻算法	77
3.1 K 近邻算法简介	77
3.2 K 近邻算法 API 初步使用	77
3.3 距离度量	77
3.4 K 值的选择	78
3.5 KNN 算法的特点及优势与劣势	79
3.6 案例	79
第 4 章 线性回归	88
4.1 线性回归简介及数学求导	88
4.2 线性回归 API 初步使用	90
4.3 线性回归的损失和优化	92
4.4 梯度下降法介绍	93

4.5	欠拟合与过拟合	94
4.6	案例	96
第5章	逻辑回归	103
5.1	逻辑回归算法简介	103
5.2	逻辑回归 API 的初步使用	107
5.3	案例及分类评估方法	108
5.4	ROC 曲线绘制	111
第6章	决策树	118
6.1	决策树算法简介	118
6.2	决策树分类原理	118
6.3	CART 剪枝	124
6.4	特征工程及特征提取	125
6.5	决策树算法 API 的初步使用	132
6.6	案例	132
第7章	支持向量机	141
7.1	SVM 算法简介及 SVM 算法 API 的初步使用	141
7.1.1	基本概念	141
7.1.2	SVM 算法 API 的初步使用	142
7.2	SVM 算法原理	144
7.3	SVM 算法的损失函数	153
7.4	SVM 算法的核函数	154
7.5	SVM 回归	155
7.6	案例	155
第8章	朴素贝叶斯分类器	160
8.1	朴素贝叶斯算法简介	160
8.2	概率知识	160
8.3	拉普拉斯估计法	161
8.4	案例	164
第9章	集成学习	178
9.1	不存在强依赖,可同时生成的并行方法	178
9.2	个体学习器有强依赖,串行生成序列方法	181
9.2.1	XGBoost 集成学习算法的工作原理	182
9.2.2	停止分裂条件判断	185
9.2.3	XGBoost 解决过拟合的问题	186

9.3 XGBoost 集成学习算法 API 初步实验	186
9.4 案例	188
第 10 章 降维算法	198
10.1 理解降维的目的和常见的方法	198
10.2 线性判别分析法	198
10.3 主成分分析法	203
10.4 案例	211
第 11 章 聚类	222
11.1 无监督学习、监督学习、半监督学习	222
11.2 聚类算法简介及聚类算法 API 函数的初步使用	223
11.3 聚类算法实现流程	224
11.4 案例	238
第 12 章 神经网络和深度学习	245
12.1 神经网络概念	245
12.2 神经网络结构、激活函数	246
12.3 神经网络求解遇到的问题	250
12.4 深度学习基础	255
参考文献	264
附录 Python 安装指南	266

绪 论

1. 人工智能的基本概念、智能的特征、人工智能的发展历史

人工智能(Artificial Intelligence, AI)是早已存在的一个研究方向,其试图让机器具备像人类一样的智能,最近 AI 的大热得益于机器学习(machine learning)的助力,尤其是深度学习(deep learning)的助力。机器学习在垃圾邮件识别、房价线性回归预测、医学辅助诊断、字符识别等领域取得了鼓舞人心的进步,而深度学习在图像识别、自然语言处理、语音识别、体育竞技等方面取得了巨大的进步。在学习本书之前,读者需充分了解人工智能、机器学习和深度学习的关系。如图 0.1 所示,机器学习是人工智能领域的子集,而深度学习是机器学习的子集。人工智能是计算机科学的一个分支(它包含多学科的交叉学科),它企图了解智能的实质,并生产出一种新的、能以与人类智能相似的方式做出反应的智能机器。人工智能研究的内容有:计算机视觉、自然语言处理、语音识别、图像识别、机器人和推荐系统等。

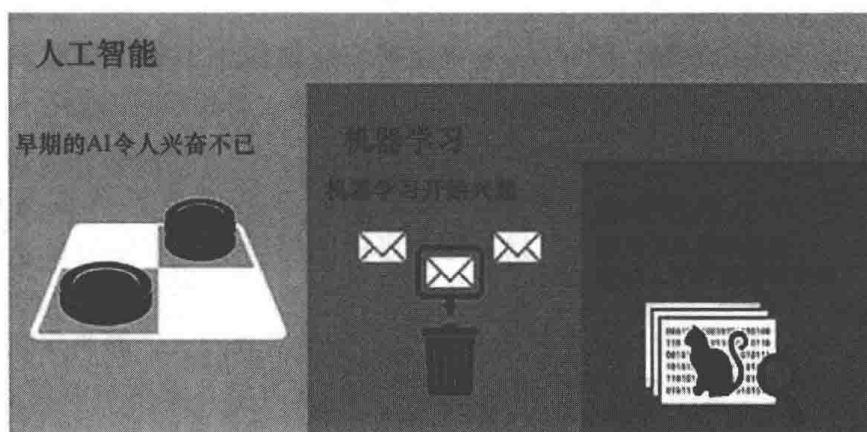


图 0.1 人工智能、机器学习和深度学习的关系

人工智能从诞生以来,理论和技术日益成熟,应用领域也不断扩大。20 世纪 50 年代,人工智能进入“推理期”,通过赋予机器逻辑推理能力使机器获得智能,当时的 AI 程序能够证明一些著名的数学定理,但由于机器缺乏知识,远不能实现真正的智能。20 世纪 70 年代,人工智能的发展进入“知识期”,即将人类的知识总结出来教给机器,使机器获得智能,在这一时期,大量的专家系统问世,在很多领域取得了大量成果,但由于人类知识量巨大,因此出现了“知识工程瓶颈”。无论是“推理期”还是“知识期”,机器都是按照人类设定的规则和总结的知识运作,因而并不“智能”,而且大量的规则和知识需要耗费的人力成本太高。

于是,一些学者试图让机器自主学习,机器学习方法应运而生,人工智能进入“机器学习时期”。2001 年,在金融贸易竞赛中,具备 AI 的机器人战胜了人类;银行使用 AI 进行金融投资和财产管理;在临床医学领域,人工神经网络被用于辅助决策、分析医学影像、分析心脏声音;在客服方面,呼叫中心的回答机器也使用 AI 技术,自动识别并答复客户的提问;在体

育竞技方面,随着 AlphaGo 的问世,曾经被认为机器人无法与人类比拼的竞技项目——围棋,也被 AI 攻克,AlphaGo 在围棋上击败了人类世界冠军李世石和柯洁,现在越来越多的职业围棋选手依赖 AI 训练自己的围棋技艺。人工智能可以对人的意识、思维过程进行模拟,虽然它不是人的智能,但却能像人那样思考,甚至有可能超过人的智能。人工智能发展史上的重要事件如图 0.2 所示。人工智能取得如此重大的进步离不开另外两个概念:机器学习和大数据。

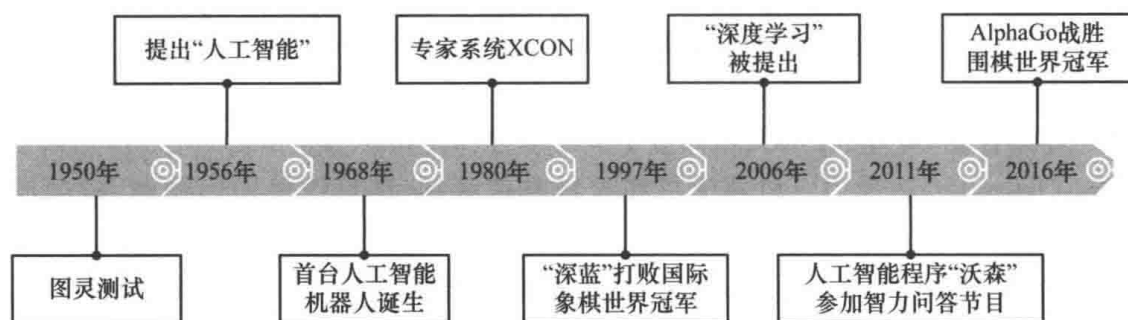


图 0.2 人工智能发展史上的重要事件

人工智能与机器学习、大数据等概念之间存在怎样的联系?研究“人工智能”的目的是让机器能够像人类一样思考并行动。目前我们离这一目标还有些遥远,而且是否应该这样做(大力研究具备人的智能的机器人)还具有争议^①。然而,实际上整个社会已经在全力推进并加速人工智能的研发进程。同时,人工智能被广泛地应用于各个领域,这被称为“应用人工智能”,即帮助人类完成各种简单任务的手段或工具。而要实现“应用人工智能”,则需要使用机器学习的技术。

关于机器学习的定义有很多, Peter Harrington 在其经典机器学习教程 *Machine Learning in Action* 中提出:“机器学习就是把无序的数据转换成有用的信息。”机器学习用大量数据训练出一个模型,然后使用该模型代替人类完成一些简单的任务,包括分类、回归和聚类分析。机器学习是目前应用人工智能较为主流的实现方式。赛事报道的新闻大同小异,可以借助于机器学习辅助撰写;医生需要机器辅助查看患者的医学影像底片,从而更准确地判断疾病的类型与程度;邮件服务器依赖机器学习鉴别垃圾邮件和正常邮件,从而有效地隔离无用的垃圾邮件……通过机器学习对数据进行训练,得出对应的算法模型,然后用这个模型分析新输入的数据,并自动生成结果,这样做可以显著地提高工作效率。

大数据(big data)提供了训练模型所需要的基础资源。Gartner Group 对大数据的定义为:需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力来适应海量、高增长率和多样化的信息资产。对这些数量惊人、形式多变的数据进行收集、管理、分析等工作的技术被称为“大数据技术”。海量数据经过预处理后变为实现应用人工智能的基础资料,计算机正是利用这些数据,并依据机器学习的算法,完成模型的训练,也即顺利完成“学习”,然后利用训练后的模型去实现人类给予的新任务。

2. 人工智能的研究内容及研究领域

人工智能的研究范围如图 0.3 所示,其中机器学习是人工智能的核心,是实现人工智能

^① 伟大的英国天体物理学家史蒂芬·霍金认为人类不应该大力研发人工智能技术,因为机器人拥有人类难以比拟的体力和记忆力,一旦机器人拥有了智能,那对人类是一种灾难。

的途径。根据学习的任务模式,机器学习可以分为四大类:监督学习、无监督学习、半监督学习和增强学习。这样分类符合麦肯锡思维的准则,即分类必须相互独立、完全穷尽。图 0.3 中的“深度学习”“集成学习”和“迁移学习”不能作为机器学习的类别,但是却属于机器学习的范畴。

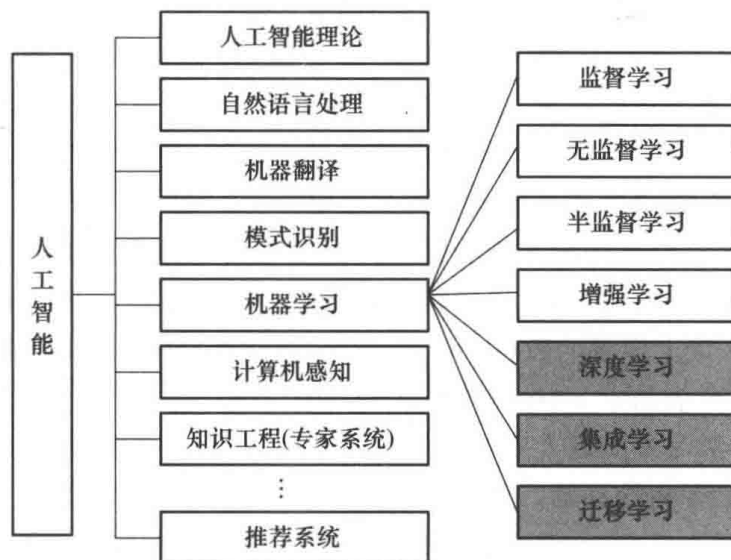


图 0.3 人工智能和机器学习的关系及机器学习的分类

监督学习(supervised learning)利用带有标签的数据来训练模型,再用训练好的模型去测试新的数据。这种学习方式类似于学生利用带有答案的习题册进行学习,然后举一反三地做习题册中没有答案的题目。在这个例子中,“答案”等价于标签,“习题”等价于数据,“习题册中没有答案的题目”等价于新的数据。在监督学习中,数据=(特征,标签),其主要任务是对数据进行分类和回归分析。

无监督学习(unsupervised learning)用于找出输入数据的模式。典型的有对数据的聚类分析,如根据电影的特征做聚类分析并进行电影的标识(属于哪类电影)。此外,无监督学习还可以用于降维,从而能够更加清晰地认识数据。在无监督学习中,数据=(特征)。

半监督学习(semi-supervised learning)介于监督学习和无监督学习之间。实际上,生活中的数据大多数没有标签,同时对大多数数据进行标记的代价是很高的,因而半监督学习适用于样例集中一部分数据没有标签的情况。在半监督学习中,数据=(特征,标签)或者(特征)。

例如,对猫和狗的照片进行分类:

- 监督学习利用已有的猫和狗的标签,对新照片进行分类;
- 无监督学习利用照片里的特征将照片聚类为猫和狗两大类;
- 半监督学习先将未标记的照片聚类生成标签,再结合已有的标签进行分类。

增强学习(reinforcement learning)是在行动中学习,与监督学习不同的是它不需要输入和标签,而是根据环境对智能体(agent)在不同状态(state)下的行为(action)进行评价。评价分为正评价(奖励)和负评价(惩罚)。在增强学习中,数据=(特征,评价)。以股票交易系统为例,股票交易市场就是环境,交易系统就是智能体,股票价格就是状态,买卖股票的动作就是行为。股票交易系统持续预测股价,如果正确则奖励,反之则惩罚,通过不断地与市

场交互,从而不断完善自己的系统,使自己越来越准确。

集成学习(ensemble learning)并不是机器学习的一个分类,它只是将若干个学习器(分类器或回归器)组合之后产生的一个新的学习器。弱分类器指那些分类准确率只稍微高于随机猜测的分类器。集成算法的优点在于保证弱分类器的多样性,而且集成不稳定的算法也能够得到比较明显的性能提升。

深度学习(deep learning)是机器学习的一个子集,它受生物学启发,通过各种神经网络来建模,通过调节各层节点的权重,建立输入与输出之间的关系。深度学习中的卷积神经网络可用于监督学习中的图片分类,循环神经网络可用于回归分析,同时,深度学习还可用于无监督学习中的聚类,在增强学习中也得到了应用。

迁移学习是将已训练好的模型参数迁移到新的模型中进行训练,而不需要从零开始学习。迁移学习的核心是找到新问题与原问题的相似性。迁移学习有一个有标签的数据 $s=(\text{特征 } s, \text{标签 } s)$ 和一个无标签的数据 $t=(\text{特征 } t)$,在这两个领域中,特征 s 和特征 t 的分布不同,迁移学习就是要借助于 s 中的知识来学习 t 中的知识。

简而言之,机器学习是实现人工智能的途径与手段,大数据则提供实现人工智能的基础资料。大数据是人工智能的基础,而使大数据转变为知识或生产力离不开机器学习,可以说**机器学习是人工智能的核心,是使机器具有类似于人的智能的根本途径**。本书将集中于机器学习的知识,通过 Python 语言(一种适用于人工智能分析的计算机高级语言)进行编程,通过实例讲解机器学习中的各个算法及其应用。开发环境可以选择:Jupyter Notebook、PyCharm 或者 Spyder 等。

第 1 章 Python 基础

Python 是一种解释型的、面向对象的、动态数据类型的高级程序设计语言。

Python 由 Guido van Rossum 于 1989 年年底发明,第一个公开发行人版发行于 1991 年。

像 Perl 语言一样,Python 源代码同样遵循 GPL(GNU General Public License) 协议。

Python 的设计具有很强的可读性,除具有其他语言经常使用的英文关键字、标点符号外,Python 还具有比其他语言更有特色的语法结构。

Python 是一种解释型语言:这意味着开发过程中没有编译环节。类似于 PHP 和 Perl 语言。

Python 是交互式语言:可以在一个 Python 提示符 `>>>` 后直接执行代码。

Python 是面向对象语言:Python 支持面向对象的编程技术。

Python 是初学者的语言:应用领域非常广泛,从简单的文字处理到 WWW 浏览器再到游戏都可以使用 Python 开发,Python 对初学者而言,其语法简单,易学易用。

1.1 开发环境、Python 基础知识

1.1.1 开发环境

本书所介绍的关于人工智能的开发环境框架是在 Windows 10 操作系统上进行搭建的。安装过程参照附录中的安装指南,基本上不会出现安装其他 IDE 时可能出现的各种路径配置等问题,搭建好以后再根据后续章节进行学习,对于初学者来说非常容易上手。

Anaconda 中的 Jupyter Notebook 相对于 PyCharm 来说,更多地用在数据探索和算法设计阶段,需要获得一段代码结果,PyCharm 更加适用于工程化的项目,如编写一个 python 库,或者编写一个项目等。对于刚上手的人来说,Jupyter Notebook 更加友好。

1.1.2 Python 基础知识

1. Python 是解释型编程语言

程序指的是一系列指令,用来告诉计算机做什么。而编写程序的关键在于,需要用计算机可以理解的语言来提供这些指令。为了有效避开所有影响给计算机传递指令的因素,计算机科学家们设计了一些符号,这些符号各有其含义,且无二义性,通常称它们为编程语言。编程语言中的每个结构都有固定的使用格式(称为语法)以及精确的含义(称为语义)。换句话说,编程语言指定了成套的规则,用来编写计算机可以理解的指令。习惯上,将这一条条指令称为计算机代码,而用编程语言来编写算法的过程称为编写代码。

Python 语言是一种编程语言,和 C、C++ 等编程语言一样,属于高级计算机语言。使

用高级计算机语言的目的是方便程序员理解和使用。但是,计算机硬件只能理解一种非常低级的编程语言,称为机器语言(二进制指令)。

使用 Python 编程语言,对两个数求和可以很自然地用 $c=a+b$ 表示,但需要设计一种方法,将高级语言翻译成计算机可以执行的机器语言,有两种方法可以实现,分别是使用编译器和使用解释器(编译型语言和解释型语言的区别如表 1.1 所示)。

表 1.1 编译型语言和解释型语言的区别

类型	原理	优点	缺点
编译型语言	通过专门的编译器,将所有源代码一次性转换成特定平台(Windows、Linux 等)执行的机器码(以可执行文件的形式存在)	编译一次后,脱离了编译器也可以运行,并且运行效率高	可移植性差,不够灵活
解释型语言	由专门的解释器,根据需要将部分源代码临时转换成特定平台执行的机器码	跨平台性好,通过不同的解释器,可将相同的源代码解释成不同平台下的机器码	一边执行一边转换,效率很低

在什么时候将源代码转换成二进制指令呢?不同的编程语言有不同的规定:

有的编程语言要求必须提前将所有源代码一次性转换成二进制指令,也就是生成一个可执行程序(Windows 下的 .exe),如 C、C++ 等,这种编程语言称为编译型语言,使用的转换工具称为编译器。

有的编程语言可以一边执行一边转换,需要哪些源代码就转换哪些源代码,不会生成可执行程序,如 Python、JavaScript、PHP、Shell、MATLAB 等,这种编程语言称为解释型语言,使用的转换工具称为解释器。

Python 属于典型的解释型语言,所以运行 Python 程序需要解释器的支持,只要在不同的平台安装了不同的解释器,代码就可以随时运行,不用担心任何兼容性问题,真正实现“一次编写,到处运行”。

Python 几乎支持所有常见的平台,如 Linux、Windows、Mac OS、Android、FreeBSD、Solaris、PocketPC 等,所写的 Python 代码无须修改就能在这些平台上正确运行。也就是说,Python 的可移植性很强。

2. Python 的优缺点

(1) Python 的优点

1) 语法简单

和传统的 C、C++ 等编程语言相比,Python 对代码格式的要求没有那么严格,这使得用户在编写代码时比较舒服,不用在细枝末节上花费太多精力。在开发 Python 程序时,用户可以专注于解决问题本身,而不用顾虑语法。

2) Python 是开源的

开源,也即开放源代码,即所有用户都可以看到源代码。官方将 Python 解释器和模块的代码开源,是希望所有 Python 用户都参与进来,一起改进 Python 的性能,弥补 Python 的不足。

3) Python 是免费的

大多数的开源软件是免费软件,Python 编程语言既开源又免费。用户使用 Python 开发或者发布自己的程序,不需要支付任何费用。

4) Python 是高级语言

Python 封装较深,隐藏了很多底层细节,如 Python 会自动管理内存(需要时自动分配,不需要时自动释放)。高级语言的优点是使用方便,缺点是隐藏了深层的设计细节。

5) Python 是解释型、跨平台语言

Python 是解释型语言,一般都是跨平台的(可移植性强)。

6) Python 是面向对象的编程语言

Python 支持面向对象,但它不强制使用面向对象。Java 是典型的面向对象的编程语言,但是它强制必须以类和对象的形式来组织代码。

7) Python 功能强大(模块众多)

Python 的模块众多,几乎可以实现所有的常见功能。除了 Python 官方提供的核心模块,很多第三方机构也会参与开发模块。

8) Python 可扩展性强

Python 的可扩展性体现在它的模块上,Python 具有脚本语言中最丰富和强大的类库,这些类库覆盖了文件 I/O、GUI(图形用户界面)、网络编程、数据库访问、文本操作等绝大部分应用场景。Python 依靠其良好的可扩展性,在一定程度上弥补了运行效率低的缺点。

(2) Python 的缺点

1) 运行速度慢

运行速度慢是解释型语言的通病。Python 速度慢不仅仅是因为一边运行一边“翻译”源代码,还因为 Python 是高级语言,屏蔽了很多底层细节。这个代价也是很大的,Python 要做很多工作,有些工作特别消耗资源,如管理内存。随着计算机的硬件速度越来越快,硬件性能的提升可以弥补运行速度慢的不足。

2) 代码加密困难

不像编译型语言的源代码会被编译成可执行程序,Python 是直接运行源代码,因此对源代码进行加密比较困难。

3. Python 的应用领域

Python 作为一种简单实用的编程语言,它的应用尤其广泛,主要应用于以下几个领域。

(1) 网页开发

Python 可以作为一种服务器语言辅助前后端开发。在前端可以帮助生成各种文件,实现网页端的动态呈现;在后端可以用来创建不同的应用程序,处理前端传过来的请求。目前 Python 有很多网页框架(Flask、Bottle、Django 等)和数据处理函数库(Pandas、NumPy、Feedparser 等)可以直接调用,能极大地提升网页开发的速度。

(2) 数据分析、数据科学

Python 有大量的数据分析和机器学习方面的模块可以使用。例如,Pandas 和 NumPy 可以搭配起来处理数据表格,SciPy 可以对数据进行统计,Matplotlib 可以生成各种图表,Scikit-Learn 模块有多种机器学习算法的函数可以调用,极大地减少了研究

者的工作量。

(3) AR/VR

AR 是指增强现实技术,VR 是指虚拟现实技术,涉及很多的视频处理、图像处理任务。而 Python 提供了相关模块,其内部具有大量的函数可以调用,如过滤图片噪声、剪切图片、调整图片灰度、旋转和调整图片大小、边缘检测、模板匹配等,都可以处理 AR/VR 所涉及的图像、视频方面的问题。

(4) 自然语言处理(NLP)

自然语言处理研究能实现人与计算机之间用自然语言进行有效通信的各种理论和方
法。自然语言处理是一门融语言学、计算机科学、数学于一体的科学。在 Python 中可以导入相关模块(如 NLTK),可以实现对语句进行分类、提取关键词、比较两个语句之间的相似度等。

(5) 网络爬虫

Python 语言很早就用来编写网络爬虫。Google 等搜索引擎公司大量地使用 Python 语言编写网络爬虫。

【例 1.1】 第一个 Python 程序。

新建一个 Python3 文件,在 In[]所在区域输入:`print ("Hello, Python!")`。界面如图 1.1 所示。

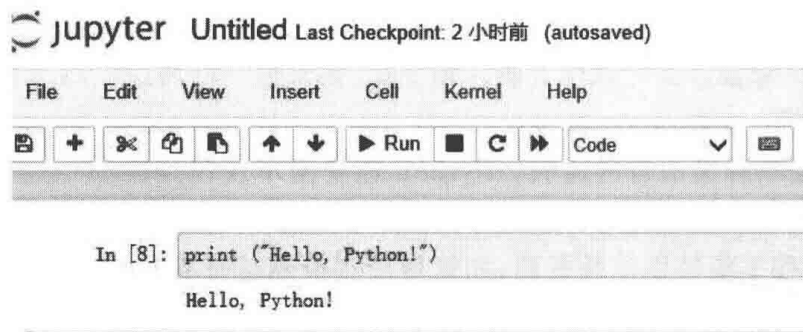


图 1.1 例 1.1 的界面

4. Python 标识符

在 Python 中,所有标识符可以包括字母、数字以及下划线,但不能以数字开头。Python 中的标识符是区分大小写的。

以下划线开头的标识符具有特殊意义。以单下划线开头的标识符(如 `_toy`)表示为受保护的成员,需通过类的接口进行访问,不能用 `from xxx import *` 来导入。以双下划线开头的标识符(如 `__foo`)代表类的私有成员。以双下划线开头和结尾的标识符代表 Python 的特殊方法专用的标识,如 `__init__()` 代表类的构造函数。

5. Python 保留字

表 1.2 列出了 Python 中的保留字,这些保留字不能用作常量或变量,或者任何其他标识符名称。所有 Python 中的保留字只包含小写字母。

表 1.2 Python 中的保留字

保留字	保留字	保留字
and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

6. 行和缩进

Python 与其他语言最大的区别就是,Python 的代码块不使用大括号{}来控制类、函数以及其他逻辑块。Python 最具特色的是用缩进来写模块,如图 1.2 所示。缩进的空格数量是可变的,但是所有代码块语句包含相同的缩进空格数量。图 1.2 中缩进为 4 个空格。

```
a=5
if a==6:
    print ("True")
else:
    print ("False")
```

False

图 1.2 用缩进来写模块

【例 1.2】 Python 的注释语句及变量的定义。

```
gender = 'female'
name = "Hilary" # 字符串
#用反斜杠(\)将一条语句写在多行
tt = name + " " + \
    gender
print(tt)
a = [0,1,2,
    3,4,5] #语句中包含 [], {} 或 () 则不需要使用多行连接符反斜杠(\)
print(a)
print(gender);print(name) #多条语句写在一行,每条语句用分号隔开
```

运行结果如图 1.3 所示。

```
Hilary female
[0, 1, 2, 3, 4, 5]
female
Hilary
```

图 1.3 例 1.2 的运行结果

关于语句和注释的说明：

- 通常一行一条语句；也有一行多条语句的，每条语句用分号隔开。
- 可以使用反斜杠(\)将一条语句写在多行；语句中包含 []、{}或()则不需要使用多行连接符反斜杠(\)。
- 可以使用引号(')、双引号(")、三引号(''或''')来表示字符串，引号的开始与结束必须相互匹配，其中三引号可以由多行组成，是编写多行文本的快捷语法，常用于文档字符串。有时，三引号用来做多行注释。
- 井号(#)作为一行注释。

Python 中的变量赋值不需要类型声明。允许同时为多个变量赋值，如 `a, b, c = 7, 9, "Hilary"`。

1.2 列表、元组、字典、集合

不同编程语言中的数据类型类似，但具体表示方法有所不同，以下是 Python 编程常用的数据类型：Number(数字)、String(字符串)、Bool(布尔型)、List(列表)、Tuple(元组)、Dictionary(字典)、Set(集合)。

数字：主要包括 int(整型)、float(浮点型)、complex(复数型)。

字符串：加了引号的字符都被认为是字符串，其声明有 3 种方式，分别是单引号、双引号和三引号。Python 中的字符串有两种数据类型，分别是 str 类型和 unicode 类型。str 类型采用 ASCII 编码，无法表示中文；unicode 类型采用 unicode 编码，能够表示任意字符，包括中文和其他语言。

布尔型：Python 的布尔类型也是用于逻辑运算，有两个值：True(真)和 False(假)。

以上几种数据类型在其他高级语言中都有介绍过，本章重点讲解 List(列表)、Tuple(元组)、Dictionary(字典)、Set(集合)等数据类型。

1. 列表

列表是 Python 中使用特别频繁的数据类型。列表可以完成大多数集合类的数据结构实现。它支持字符、数字、字符串，甚至可以包含列表(即嵌套)。列表用 [] 标识，是 Python 通用的复合数据类型。创建列表使用 []，如果要创建一个空的列表，直接用 [] 表示即可。

列表中值的切割需要用到变量 [上界:下界]，可以截取相应的列表，从左到右索引默认从 0 开始，从右到左索引默认从 -1 开始，下标可以为空，表示取到头或尾。

列表的函数和方法分别如表 1.3 和表 1.4 所示。