

第 1 章

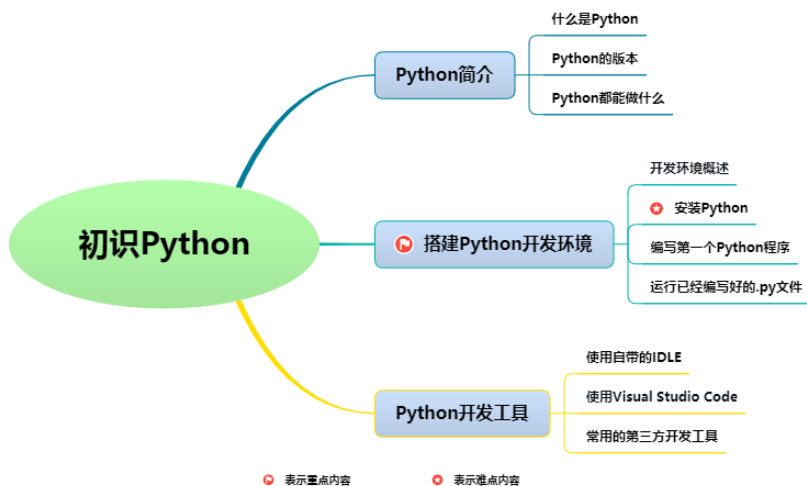


初识 Python

Python 是一种跨平台的、开源的、免费的、解释型的高级编程语言。近几年 Python 发展势头迅猛，在 2022 年 11 月的 TIOBE 编程语言排行榜中已经晋升到第 1 名，而在 IEEE Spectrum 发布的 2021 年度编程语言排行榜中，Python 连续 5 年夺冠。另外，Python 的应用领域非常广泛，如 Web 编程、图形处理、黑客编程、大数据处理、网络爬虫和科学计算等，Python 都可以实现。

作为 Python 开发的起步，本章将先对学习 Python 需要了解的一些基础内容进行简要介绍，然后重点介绍如何搭建 Python 开发环境，最后介绍常见的几种 Python 的开发工具。

本章知识架构及重难点如下。



1.1 Python 简介



1.1.1 什么是 Python

Python（发音[ˈpaɪθən]）本义是指“蟒蛇”（这里需要说明的是，Python 并不是以蟒蛇命名，而是以电视节目 Monty Python’s Flying Circus 来命名的），标志如图 1.1 所示。它的设计哲学为优雅、明确、简单。实际上，Python 也是按照这个理念做的，以至于现在网络上流传着“人生苦短，我用 Python”的说法。可见 Python 有着简单、开发速度快、节省时间和精力等特点。

Python 本身并非所有的特性和功能都集成到语言核心，而是被设计为可扩充的。它具有丰富和强

大的库，能够把用其他语言（尤其是 C/C++）制作的各种模块很轻松地联结在一起。为此，Python 常被称为“胶水”语言。

在 1991 年 Python 的第一个公开发行人问世之后，Python 的发展并不突出。自从 2004 年以后，Python 的使用率呈线性增长。在 2010 年时，Python 赢得 TIOBE 2010 年度语言大奖。直到 2021 年，IEEE Spectrum 发布的年度编程语言排行榜中，Python 已经连续 5 年夺冠，如图 1.2 所示。



图 1.1 Python 标志

Rank	Language	Type	Score
1	Python	🌐 🖥️ ⚙️	100.0
2	Java	🌐 📱 🖥️	95.4
3	C	📱 🖥️ ⚙️	94.7
4	C++	📱 🖥️ ⚙️	92.4
5	JavaScript	🌐	88.1
6	C#	🌐 📱 🖥️ ⚙️	82.4
7	R	🖥️	81.7
8	Go	🌐 🖥️	77.7
9	HTML	🌐	75.4
10	Swift	📱 🖥️	70.4

图 1.2 2021 年度编程语言排行榜前 10 名

1.1.2 Python 的版本

Python 自发布以来，主要经历了 3 个版本，分别是 1994 年发布的 Python 1.0 版本（已过时），2000 年发布的 Python 2.0 版本（已停止更新）和 2008 年发布的 Python 3.0 版本（现已更新到 3.11.x）。



说明

Python 版本更新较快，几乎两个月就升级一次，这也导致很多扩展库的发行总是滞后于 Python 的发行版本。因此，在选择 Python 时，一定要先考虑清楚自己的学习目的。例如，打算做哪方面的开发，需要用到哪些扩展库，以及扩展库支持的最高 Python 版本等。明确这些问题后，再做出选择。

1.1.3 Python 能做什么

Python 是一种功能强大，并且简单易学的编程语言，因而广受好评，那么 Python 能做什么呢？概括起来有以下几个方面。

1. Web 开发

使用 Python 的一个基本应用就是进行 Web 开发。在国内，大一些的使用 Python 做基础设施的公司有豆瓣、知乎、美团、饿了么，以及搜狐等。在国外，Google 在其网络搜索系统中广泛应用了 Python，并且聘用了 Python 之父。另外，YouTube 视频分享服务大部分也是用 Python 编写的，如图 1.3 所示。



图 1.3 Web 开发应用 Python 的公司

2. 大数据处理

随着近几年大数据的兴起，Python 也得到了前所未有的爆发。Python 借助第三方的大数据处理框架可以很容易地开发出大数据处理平台。到目前为止，Python 是金融分析、量化交易领域里使用最多的语言之一。例如，美国银行就利用 Python 语言开发出了新产品和基础设施接口，用于处理金融数据。

3. 人工智能

人工智能（artificial intelligence），英文缩写为 AI。Python 之所以这么火，主要是借助于人工智能的发展。Python 是一种脚本语言，它更适合做人工智能领域，因为在人工智能领域使用 Python 比其他编程语言具有更大的优势。主要的优势在于，它简单、快速、可扩展（主要体现在可以应用多个优秀的人工智能框架）等。另外，Python 中的机器学习可以实现人工智能领域中的大多数需求。

4. 自动化运维开发

掌握一种开发语言已经成为高级运维工程师的必备技能。Python 是一种简单、易学的脚本语言，它能满足绝大部分自动化运维的需求。对于通常不会开发的运维工程师来说，想学习一种开发语言，Python 则是首选。

5. 云计算

Python 可以广泛地在科学计算领域发挥独特的作用。Python 通过强大的支持模块可以在计算大型数据、矢量分析、神经网络等方面高效率地完成工作，尤其是在教育科研方面，可以发挥出独特的优势。从 1997 年开始，NASA 就在大量使用 Python 进行各种复杂的科学运算。现在终于发明了一套云计算软件，取名为 OpenStack（开放协议栈），并且对外公开发布。

6. 网络爬虫

网络爬虫（也称为 spider）始于也发展于百度、谷歌。但随着近几年大数据的兴起，爬虫应用被提升到前所未有的高度。多数分析挖掘公司都以爬虫的方式得到不同来源的数据集合，最后为其所用，进而构建属于自己的大数据综合平台。在爬虫领域，Python 几乎是霸主地位，通过它提供的标准支持库基本上可以做到随意获取想要的数据库。

7. 游戏开发

通过 Python 完全可以编写出非常棒的游戏程序，例如，知名的游戏《文明 6》就是用 Python 编写的。另外，在网络游戏开发中 Python 也有很多应用。它作为游戏脚本被内嵌在游戏中，这样做的好处

是既可以利用游戏引擎的高性能，又可以受益于脚本化开发等优点。



说明

Python 的应用领域远比上面提到的多得多。例如，使用 Python 对图形/图像进行处理、编程控制机器人、数据库编程、编写可移植的维护操作系统的工具，以及进行自然语言分析等。

1.2 搭建 Python 开发环境

1.2.1 开发环境概述



所谓“工欲善其事，必先利其器”。在正式学习 Python 开发前，需要先搭建 Python 开发环境。由于 Python 是跨平台的，因此可以在多个操作系统上进行编程，并且编写好的程序可以在不同系统上运行。常用的操作系统及其说明如表 1.1 所示。

表 1.1 进行 Python 开发常用的操作系统及其说明

操 作 系 统	说 明
Windows	推荐使用 Windows 10。另外，Python 3.9 及以上版本不能在 Windows 7 系统上使用
Mac OS	从 Mac OS X 10.3 (Panther) 开始已经包含 Python
Linux	推荐 Ubuntu 版本



说明

在个人开发学习阶段推荐使用 Windows 操作系统。本书采用的就是 Windows 操作系统。

1.2.2 安装 Python



要进行 Python 开发，需要先安装 Python 解释器。因为 Python 是解释型编程语言，所以需要一个解释器，这样才能运行我们写的代码。这里说的安装 Python 实际上就是安装 Python 解释器。下面将以 Windows 操作系统为例介绍如何安装 Python。

1. 下载 Python 安装包

在 Python 的官方网站中，可以很方便地下载 Python 的开发环境，具体下载步骤如下。

(1) 打开浏览器（如 Google Chrome 浏览器），进入 Python 官方网站，地址是 <https://www.python.org/>，如图 1.4 所示。



说明

如果选择 Windows 菜单项时，没有显示右侧的下载按钮，则应该是页面没有加载完全，在加载完成后就会显示，请耐心等待。

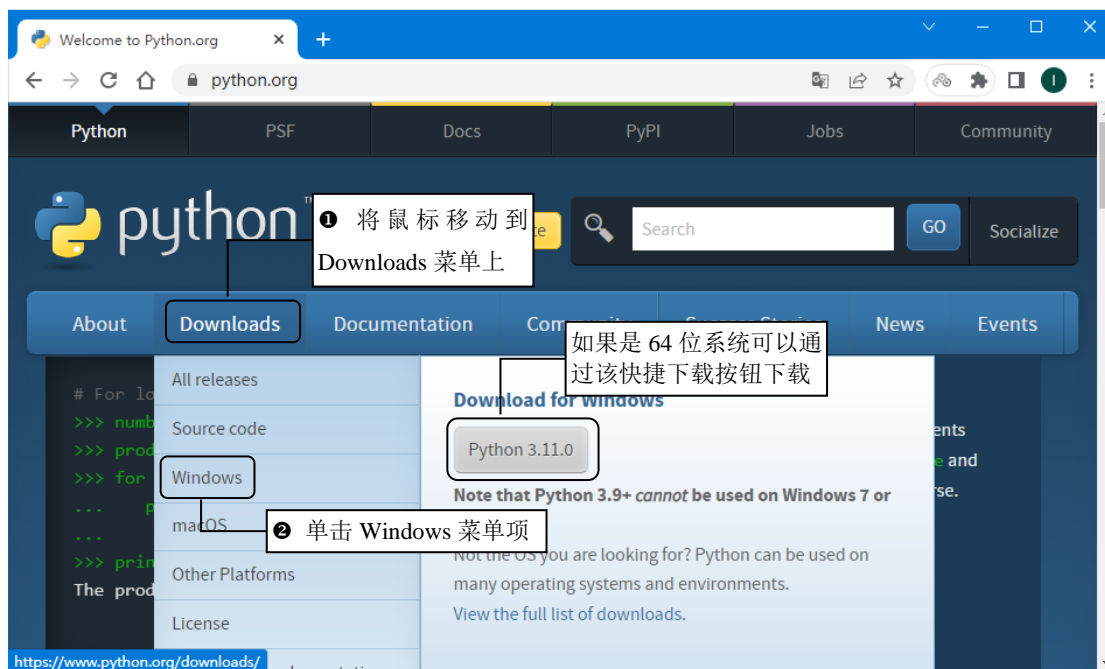


图 1.4 Python 官方网站首页

(2) 将鼠标移动到 Downloads 菜单上，将显示与下载有关的菜单项。如果使用的是 64 位的 Windows 操作系统，那么直接单击 Python 3.11.x 按钮下载 64 位的安装包；否则，单击 Windows 菜单项，进入详细的下载列表中。在下载列表中，将列出 Python 不同版本的下载连接，读者可以根据需要下载。这里单击 Windows 菜单项，进入如图 1.5 所示的下载列表。

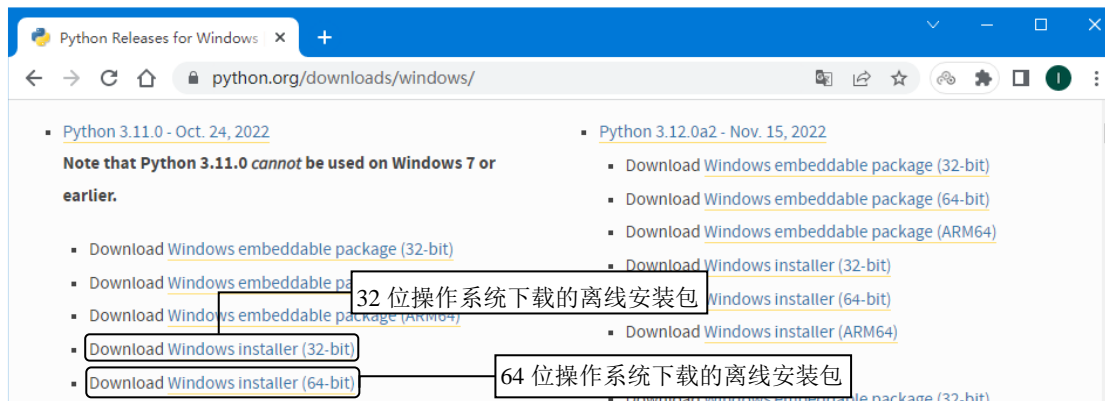


图 1.5 适合 Windows 系统的 Python 下载列表



说明

在如图 1.5 所示的列表中，带(32-bit)的，表示是在 Windows 32 位操作系统上使用的；而带(64-bit)的，则表示是在 Windows 64 位操作系统上使用的。另外，标记为 embeddable package 的，表示嵌入式安装；标记为 installer 的，表示通过可执行文件 (*.exe) 方式离线安装；标记为 embeddable zip file 的，表示嵌入式版本，可以集成到其他应用中。

(3) 在 Python 下载列表页面中, 将列出 Python 提供的各个版本的下载链接。读者可以根据需要下载。当前 Python 3.x 的最新稳定版本是 3.11.0, 所以找到如图 1.5 所示的位置, 单击 Download Windows installer (64-bit) 超链接, 下载适用于 Windows 64 位操作系统的离线安装包, 如图 1.6 所示。

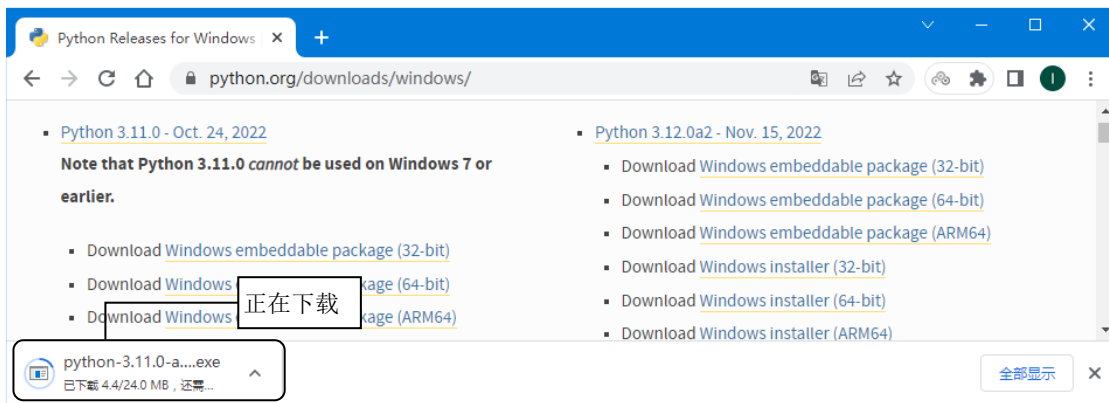


图 1.6 正在下载 Python



注意

Python 3.11.x 版本需要在 Windows 10 及以上操作系统上安装, 如果您的操作系统是 Windows 10 以下版本, 那么可以在下载列表页面中下载 Python 3.8.x 版本。

(4) 下载完成后, 浏览器会自动提示“此类型的文件可能会损害您的计算机。您仍然要保留 python-3.11.0-amd64.exe 吗?”, 此时, 单击“保留”按钮, 保留该文件即可。

(5) 下载完成后, 将得到一个名称为 python-3.11.0-amd64.exe 的安装文件。

2. Windows 64 位系统上安装 Python

在 Windows 系统上安装 Python 3.x 的步骤如下。

(1) 双击安装文件 python-3.11.0-amd64.exe, 将显示安装向导对话框, 选中 Add Python.exe to PATH 复选框, 表示将自动配置环境变量, 如图 1.7 所示。



图 1.7 Python 安装向导

**误区警示**

一定要选中 Add Python to PATH 复选框，否则在后面学习中会出现“XXX 不是内部或外部命令”的错误。

(2) 单击 Customize installation 按钮，进行自定义安装（自定义安装可以修改安装路径），这里采用默认设置，如图 1.8 所示。

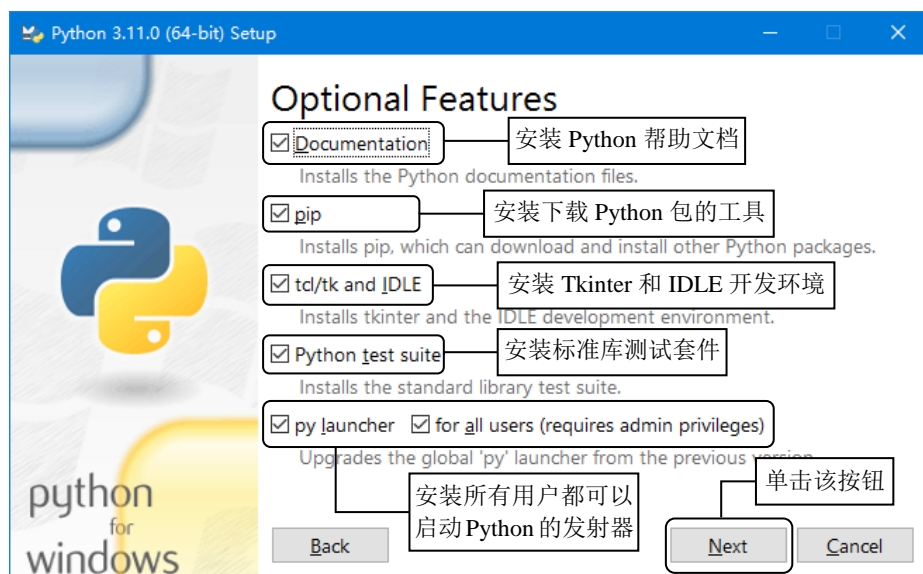


图 1.8 设置要安装选项对话框

(3) 单击 Next 按钮，在打开的高级选项对话框中，设置安装路径为 C:\Python\Python311，其他

采用默认设置，如图 1.9 所示。

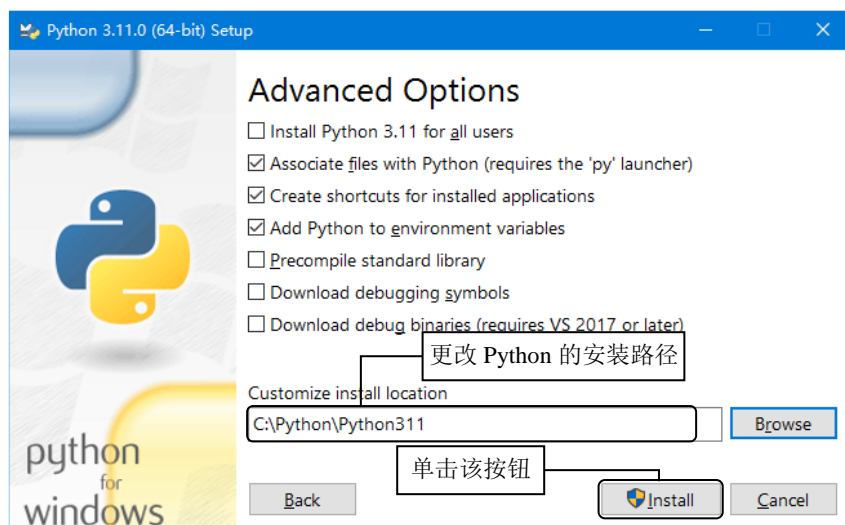


图 1.9 高级选项对话框

(4) 单击 **Install** 按钮，将开始安装 Python，并且显示安装进度。在安装完成后，将显示如图 1.10 所示的对话框。

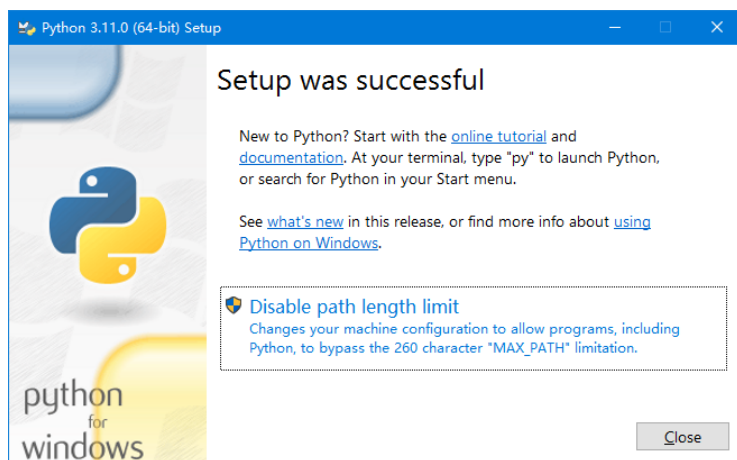


图 1.10 安装完成对话框



误区警示

安装 Python 时，有时可能会弹出如图 1.11 所示的错误提示对话框，并且不能再继续安装。

出现该错误的原因是权限设置的问题。解决方法是，在安装包文件上右击，在弹出的快捷菜单中选择“以管理员身份运行”菜单项，再继续安装即可。

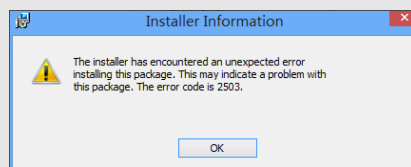


图 1.11 出现 2503 错误提示框

3. 测试 Python 是否安装成功

Python 安装成功后，需要检测 Python 是否真的安装成功。例如，在 Windows 10 系统中检测 Python 是否真的安装成功，可以在“开始”菜单右侧的“在这里输入你要搜索的内容”文本框中输入 cmd 命令，然后按 Enter 键，启动“命令提示符”窗口，再在当前的命令提示符后面输入 python，并按 Enter 键，如果出现如图 1.12 所示的信息，则说明 Python 已安装成功，同时也进入交互式 Python 解释器中。

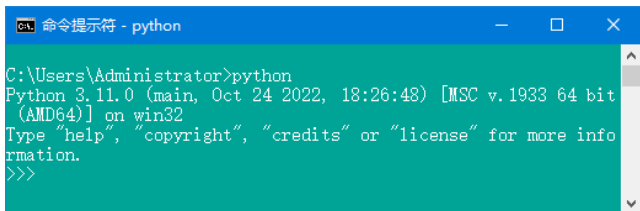


图 1.12 在“命令提示符”窗口中运行的 Python 解释器

图 1.12 中的信息是笔者计算机中安装的 Python 的相关信息，其中包括 Python 的版本、该版本发行的时间、安装包的类型等信息。所以如果与此信息不完全相同也没关系，只要命令提示符变为>>>，就说明 Python 已经准备就绪，正在等待用户输入 Python 命令。这也表示 Python 已安装成功。



如果输入 python 后，没有出现如图 1.12 所示的信息，而是显示“‘python’不是内部或外部命令，也不是可运行的程序或批处理文件。”，那么需要在环境变量中配置 Python。

1.2.3 编写第一个 Python 程序



作为程序开发人员，学习新语言的第一步就是输出 Hello World。学习 Python 开发也不例外，我们也是从 Hello World 开始。在 Python 中，可以通过以下两种方法编写 Hello World 程序。

1. 在“命令提示符”窗口启动的 Python 解释器中实现

【例 1.1】在“命令提示符”窗口中启动的 Python 解释器中实现第一个 Python 程序。（实例位置：资源包\TM\sl\01\01）

在“命令提示符”窗口中启动的 Python 解释器中编写 Python 程序非常简单方便，下面是编写第一个程序 Hello World 的具体步骤。

（1）在“开始”菜单右侧的“在这里输入你要搜索的内容”文本框中输入 cmd 命令，并按 Enter 键，启动“命令提示符”窗口，然后在当前的 Python 提示符后面输入 python，并按 Enter 键，进入 Python 解释器中。

（2）在当前的 Python 提示符>>>的右侧输入以下代码，并按 Enter 键：

```
print("Hello World")
```



在上述代码中，一对小括号()和双引号""都必须在英文（即半角）状态下输入，并且 print 全部为小写字母。因为 Python 的语法是区分字母大小写的。

运行结果如图 1.13 所示。

2. 在 Python 自带的 IDLE 中实现

通过例 1.1 可以看出，在“命令提示符”窗口的 Python 解释器中，编写 Python 代码时，代码颜色是纯色的，不方便阅读。实际上，在安装 Python 时，会自动安装一个开发工具 IDLE，通过它编写 Python 代码时，将会用不同的颜色显示代码。这样代码将更容易阅读。下面将通过一个具体的例子演示如何打开 IDLE，并编写 Hello World 程序。

【例 1.2】在 IDLE 中实现第一个 Python 程序。（实例位置：资源包\TM\sl\01\02）

在 Python 自带的 IDLE 中编写 Python 程序同样非常简单方便，下面是编写第一个程序 Hello World 的具体步骤。

- （1）单击 Windows 10 系统的“开始”菜单，然后依次选择“所有程序”→Python 3.11→IDLE (Python 3.11 64-bit) 菜单项，即可打开 IDLE 窗口，如图 1.14 所示。
- （2）在当前的 Python 提示符>>>的右侧输入以下代码，并按 Enter 键：

```
print("Hello World")
```

运行结果如图 1.15 所示。

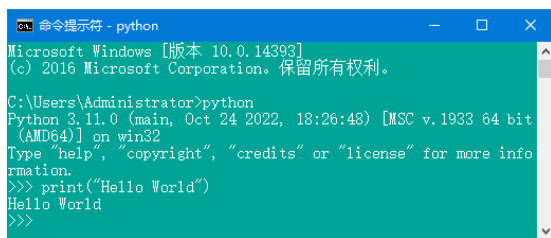


图 1.13 在“命令提示符”窗口中输出 Hello World

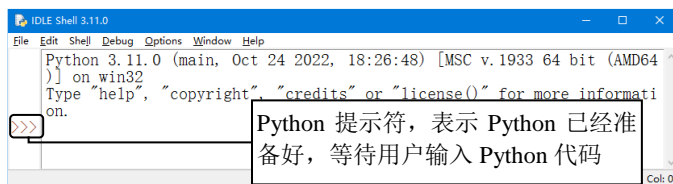


图 1.14 IDLE 窗口

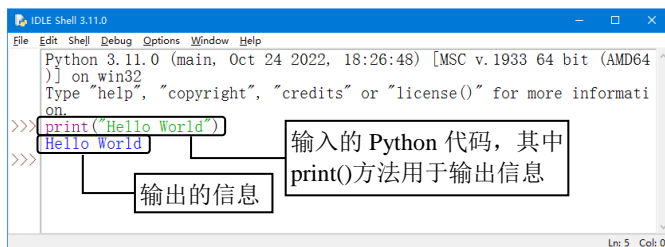


图 1.15 在 IDLE 中输出 Hello World



误区警示

如果在中文（即全角）状态下输入代码中的小括号()或者双引号""，那么将产生语法错误。例如，在 IDLE 开发环境中输入下列代码（其中括号()和双引号""在中文状态下输入）：

```
print（“Hello World”）
```

按 Enter 键运行后，将会出现如图 1.16 所示的错误提示。

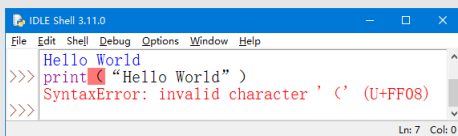


图 1.16 在中文状态下输入小括号和双引号时出现的错误

1.2.4 运行已经编写好的.py 文件



在 1.2.3 节中已经介绍了如何在 Python 交互模式中直接编写并运行 Python 代码。这里，如果已经编写好一个.py 的 Python 文件，那么应该如何运行它呢？

要运行一个已经编写好的.py 文件，可以在“开始”菜单右侧的“在这里输入你要搜索的内容”文本框中输入 cmd 命令，并按 Enter 键，启动“命令提示符”窗口，然后按照以下格式输入代码：

```
python 完整的文件名（包括路径）
```

例如，要运行 D:\demo.py 文件，可以使用以下代码：

```
python D:\demo.py
```

运行结果如图 1.17 所示。



说明

在运行.py 文件时，如果文件名或者路径比较长，可先在“命令提示符”窗口中输入 python 加一个空格，然后直接把文件拖曳到空格的位置上，这时文件的完整路径将显示在空格的右侧，再按 Enter 键运行。

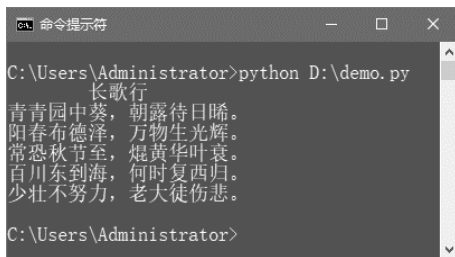


图 1.17 在 Python 交互模式下运行.py 文件