



第1章



Python 语言基础

Python 是一种高层次的具有解释性、编译性、互动性的面向对象的脚本语言。

由于 Python 编程易学易用，且积累了大量的工具库、架构，开发效率相比其他语言更高。而人工智能涉及大量的数据计算和编程设计，因此，Python 不仅成为人工智能技术的首选编程语言，而且也可成为初学者学习程序设计的首选语言。

本章将通过“中国诗词测试游戏编程”项目范例，进行自主、协作、探究式学习，让同学们认识 Python 平台，理解 Python 语言的三种基本结构及 Python 的文件操作，学会使用平台编程解决简单问题，从而将知识建构、技能培养与思维发展融入运用数字化工具解决问题和完成任务的过程中，促进计算思维的发展，完成项目学习目标。

- ◎ Python 平台及其使用
- ◎ Python 程序设计基础知识
- ◎ 顺序结构程序设计
- ◎ 选择结构程序设计
- ◎ 循环结构程序设计
- ◎ Python 的文件操作



项目范例：中国诗词测试游戏编程

● 情境

高中生小龙想学习 Python 编程，同时想用它来做一些有趣的事。他在思考，编程可以解决自己学习和生活中的哪些问题呢？是用于辅助计算，用来验证某种猜想，还是用来模拟生活中的一些现象，又或是进行自动化的控制呢？他灵光一现，编程这么“现代化”的事情，要是能用它开展一场“古今对话”，不是很有趣吗？正好，文学、历史等文化知识的学习是他每天都要面对的事情。于是，他想用 Python 来编写有关文化知识的测试游戏，这样在学习编程的同时，又可以回顾文化知识。“文化+编程”之旅一举多得，太棒啦！

● 主题

中国诗词测试游戏编程。

● 规划

根据项目范例的主题，在小组中进行讨论，利用思维导图工具，制订项目学习规划，如图 1-1 所示。

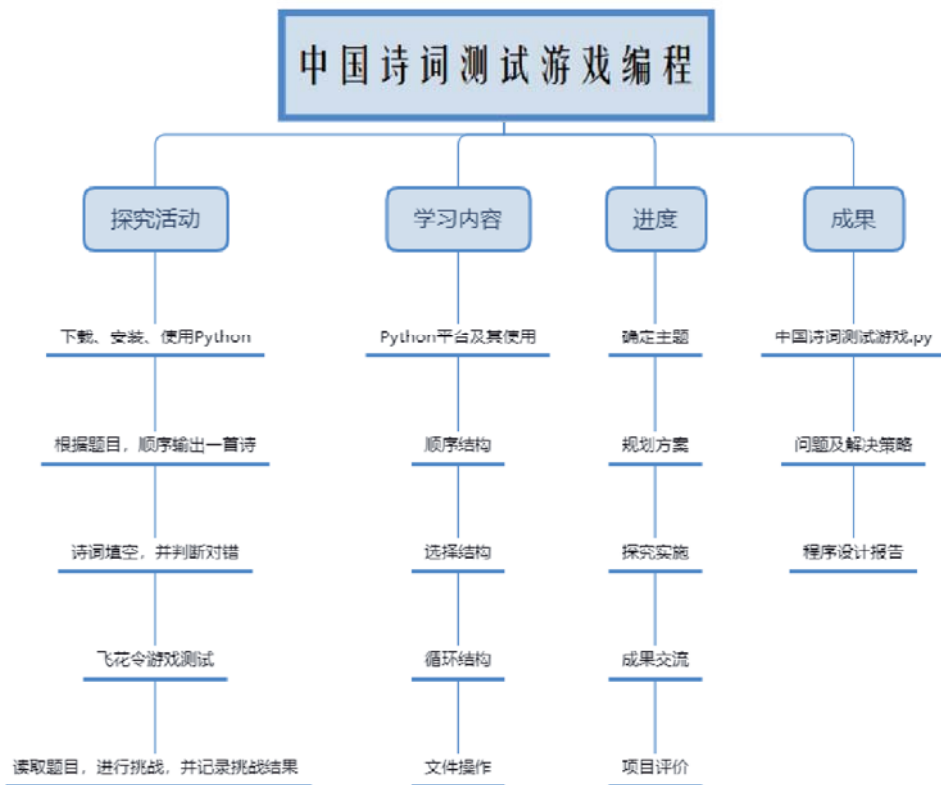


图 1-1 “中国诗词测试游戏编程”项目学习规划

● 探究

根据项目学习规划的安排，通过调查和案例分析、文献阅读或网上搜索资料，开展“中国诗词测试游戏编程”项目学习探究活动，如表 1-1 所示。

表 1-1 “中国诗词测试游戏编程”项目学习探究活动

探究活动	学习内容	知识技能
Python 平台及其使用	Python 特点	了解 Python 语言的产生及其基本特点； 熟悉 Python 平台的下载、安装与使用
	Python 平台	
	平台使用	
Python 程序设计基础知识	常量与变量	了解变量、常量、数据类型、函数等程序设计基础知识
	数据类型	
	运算符与表达式	
	函数	
顺序结构	顺序结构程序	熟悉输入、赋值与输出语句的使用； 熟练使用顺序结构编程
选择结构	条件语句	了解 Python 条件语句及其嵌套； 熟练使用 Python 条件语句编程
	条件语句嵌套	
循环结构	for 循环	了解 Python 循环语句的两种格式； 熟练使用 Python 循环语句编程
	while 循环	
文件操作	打开文件函数	掌握 Python 文件操作的基本方法； 熟练利用读、写函数编写程序
	读、写函数	

● 实施

实施项目学习各项探究活动，认识 Python 的产生及其特征，理解利用三种程序结构进行程序设计的意义和作用，并利用三种程序结构编写 Python 程序，实现项目主题。

● 成果

在小组开展项目范例学习的过程中，利用思维导图工具梳理小组成员在“头脑风暴”活动中的观点，建立观点结构图，运用多媒体创作工具（如演示文稿、在线编辑工具等），综合加工和表达，形成如图 1-2 所示的可视化学习成果，并通过各种分享平台发布可视化学习成果以及诗词测试的游戏程序。



图 1-2 “中国诗词测试游戏编程”可视化报告目录截图

● 评价

根据本书附录中的“项目活动评价表”对项目范例的学习过程和学习成果，在小组和全班中，或在网络上开展交流，进行自评和互评。

● 项目选题

请同学们以 3 ~ 6 人组成一个小组，从下面选择一个参考主题，或者自拟一个感兴趣的主题，开展小组项目学习：

1. “有趣的数学”计算游戏编程。
2. “有趣的物理”验证游戏编程。
3. “有趣的电器”模拟控制游戏编程。

● 项目规划

各小组根据本组的项目选题，参照项目范例的样式，利用思维导图工具，制定相应的项目规划和实施方案。

● 方案交流

各小组将完成的方案在全班中进行展示交流，师生共同探讨、完善项目方案。

第1节 Python 平台及其使用

● 小龙的想法

小龙想用 Python 开展一场编程与诗词的“碰撞”，当然要先去了解 Python，熟悉如何使用 Python。于是他开始了探究学习的第一步——Python 平台的产生、特点，以及安装和使用。

Python 不仅受到很多专业人士的青睐，也是一种适合初学者学习和使用的程序设计语言。由于 Python 所支持的库非常丰富，因而使用 Python 编程可以解决各种各样的实际问题。

一、Python 的产生及其特点

1. Python 的产生

Python 是由吉多·范罗苏姆（见图 1-3）于 1989 年开发的程序设计语言，并于 1991 年发布了第一个公开发行人版本。近些年，在编程排行榜上，Python 一直名列前茅。



图 1-3 吉多·范罗苏姆（Guido van Rossum）

Python 的产生当然有它的时代需求，在此希望大家重点关注它“好（hǎo）玩”的特点。

Python 译作“蟒蛇”，是不是感觉很强大？事实上，Guido 是 BBC 喜剧电视剧——蒙提·派森的飞行马戏团（Monty Python's Flying Circus）的爱好者。1989 年的圣诞节，在阿姆斯特丹，为了打发时间，Guido 决定开发一个新的脚本解释程序，并取名为“Python”。希望所有人都能



用一颗“好(hào)玩”的心来学习编程，因为 Python 可以做很多好玩的事。

2. Python 的特点

编程的语言那么多，为什么要选择 Python 呢？在它的众多优点中，以下几个格外吸引人：

(1) Python 是开源的。“开源”意味着用户可以扩展 Python，创建一些新的工具供自己和其他人使用。所以，你可以找到很多免费的 Python 工具下载并使用。

(2) Python 兼容性强。Python 可以在 Windows、苹果等各种类型的计算机上运行，如 UNIX、Windows、MacOS、Ubuntu、OS/2 等。

(3) Python 是面向对象的编程语言。比如函数、模块、字符串、数字等都是对象。Python 支持类和多层继承等的面向对象编程技术。

(4) Python 简单、易读。Python 是一种代表简单主义思想的编程语言。了解了它的语法规则后，阅读 Python 程序就像在读英语一样流畅。

(5) Python 很强大。Python 有庞大的标准库，以及丰富的、功能齐全的其他第三方库，可以帮助用户处理各种工作。

(6) Python 很优雅。经过 Python 设计者以及 Python 社区文化的长期熏陶，逐渐演化出一种独特的、优雅的编程风格。好的 Python 代码简练、明确、优雅、执行效率高，可以说它很“pythonic”。

为了让大家对“pythonic”有更清晰的了解，下面举两个例子。

(1) 解决数学问题是编程的一大应用，如果要计算“ $0^2+1^2+2^2+3^2+4^2+5^2+6^2+7^2+8^2+9^2=?$ ”，只要运行程序 1-1，就可以得到如图 1-4 所示的结果，代码只有 1 行！

★程序 1-1 自然数的平方和

```
print (sum(i**2 for i in range(10)))
```

```
>>> print (sum(i**2 for i in range(10)))
285
>>>
```

得到结果 输入命令

图 1-4 求自然数的平方和程序及其结果

(2) 字符串操作在编程时较为常见，对于已有的几个字符串，想要通过多种方式（如+、-、*、/、逗号等）将它们连接成一串，使用优雅的 Python，只需 2 行代码！

运行程序 1-2，得到的结果如图 1-5 所示。

★程序 1-2 唐朝诗人

```
names = ['李白', '杜甫', '白居易', '王维', '李商隐', '杜牧', '刘禹锡', '王昌龄', '岑参', '李贺']
print ('- '.join(names))
```

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> names = ['李白', '杜甫', '白居易', '王维', '李商隐', '杜牧', '刘禹锡', '王昌龄', '岑参', '李贺']
>>> print(' - '.join(names))
李白 - 杜甫 - 白居易 - 王维 - 李商隐 - 杜牧 - 刘禹锡 - 王昌龄 - 岑参 - 李贺
>>> |
```

图 1-5 字符串连接程序及其运行结果

【交流】

上网搜索或到图书馆查阅资料，了解“Python 的优雅”，并在小组内进行交流。

二、Python 的下载与安装

Python 是跨系统的程序设计语言，关于 Python 的下载与安装，这里以在 Windows 系统平台的安装为例介绍其过程，其他系统的使用方法大同小异。

1. Python 的下载

搜索 Python，进入 Python 官方网站的下载(Downloads)页面，根据操作系统，找到合适的版本，如图 1-6 所示，自行下载解压。（本书所有示例程序都是基于 Python 3 系列版本）



图 1-6 Python 官网下载页面

2. Python 的安装

为了方便使用，安装 Python 时，注意不要选择默认安装，选择自定义安装比较好。具体操作如下。

(1) 在合适位置建立安装文件夹，如 D:\Python。

(2) 运行安装文件（如 python-3.7.0-amd64.exe），在弹出的安装界面中选择 Customize Installation（定制安装），并注意勾选 Add Python 3.7 to PATH（添加 Python 3.7 到路径）。

(3) 在下一个页面，注意勾选 Precompile standard library（预编译标准库），并通过 Browse（浏览）按钮选择创建好的安装文件夹（如 D:\Python），然后单击 Install（安装）按钮，一会儿就可完成安装。

(4) 安装完后，可在桌面创建一个 Python 的快捷方式，其路径和文件类似 D:\Python\Lib\

idlelib\idle.pyw。

注意：如果在安装 Python 时选择的是默认安装 (Now Installation)，那么，就必须在安装完后，手工配置环境变量（“控制面板\系统和安全\系统”-“高级系统设置”-“环境变量”-编辑系统变量“path”-“新建”-Python 所在路径……）和创建快捷方式等。

【体验】

- (1) 尝试下载 Python 新的发行版本，并进行安装、调试。
- (2) Python 可以将其他语言编写的控件或库文件变为扩展库而为己所用。上网搜索或到图书馆查阅资料，体验如何下载安装扩展包。

三、Python 平台的使用

Python 平台可选择以下三种模式（交互会话、程序文件、在线练习器）之一使用。

1. 交互会话模式

在 Windows 系统中安装完 Python 后，运行 Python 即进入交互会话模式，操作如图 1-7 所示。

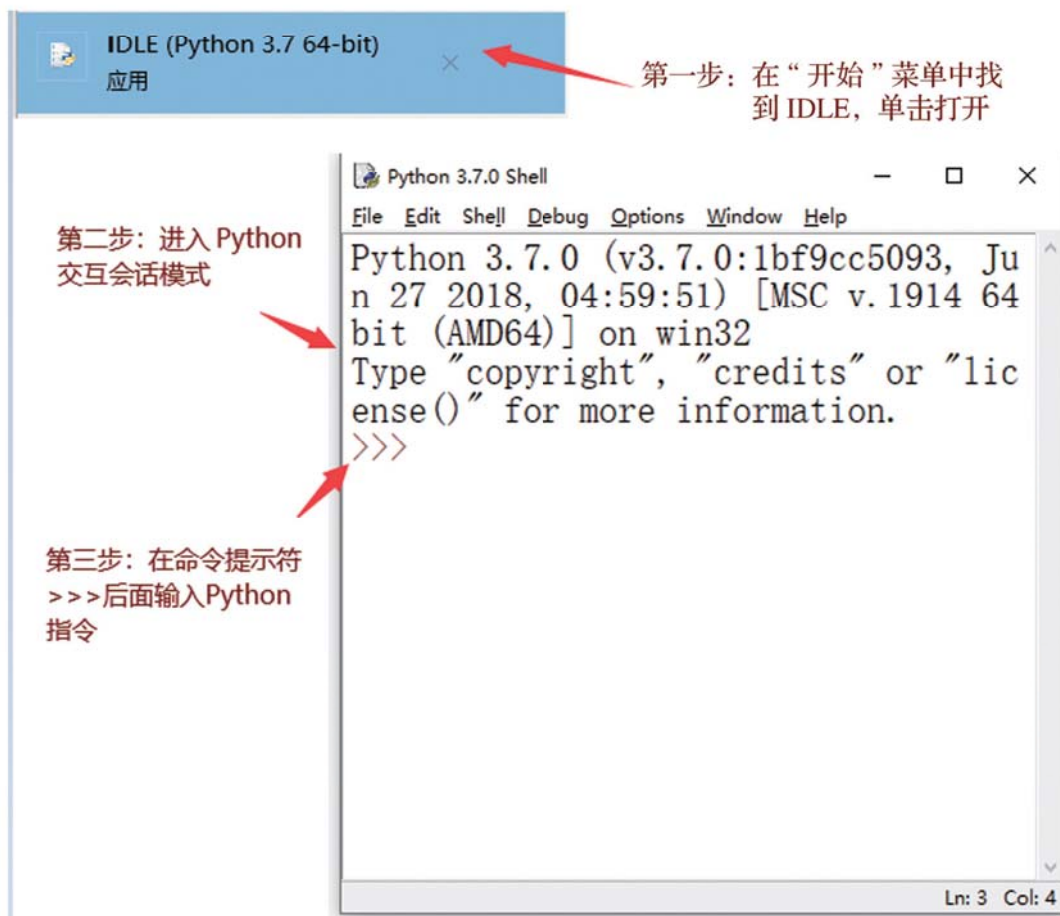


图 1-7 Python 交互会话模式

Python 交互会话模式有以下几点需要注意。

(1) 只能输入 Python 指令。

在 Python 交互模式下输入 Python 代码指令，如 `print(2**3)` 或 `"李白"*5` 等，按 Enter 键结束命令，运行结果将如图 1-8 所示显示在 Python Shell 窗口（就是启动 IDLE 时出现的那个窗口）中。

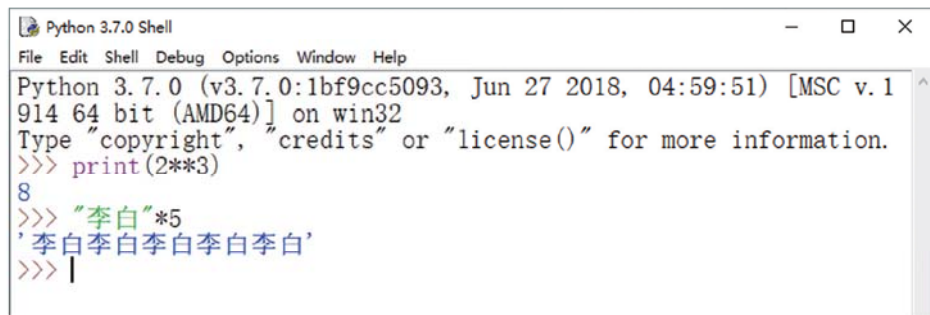


图 1-8 Python Shell 窗口

(2) 打印语句可以省略关键词。

在交互模式下不需要输入完整的打印语句，解释器能够自动打印表达式的结果，但是在程序文件模式中则需要写出完整的 `print()` 语句才能打印结果。

(3) 提示符的变换和复合语句。

当在交互模式下输入两行或多行复合语句（如 `if` 语句、`for` 循环语句等）时，提示符会由 `>>>` 变成 `...`；如果要结束复合语句的输入并执行它，则必须按两次 Enter 键。

(4) 一次运行一条语句。

若想测试某一条指令，交互模式是一个很好的测试方法。输入指令然后按 Enter 键即可看到执行结果，非常方便。

【实践】

在 Python 交互会话模式下输入 Python 代码，显示“Hello World!”。

2. 程序文件模式

程序文件模式下，需要重点掌握四个内容：打开、新建、保存和运行。

1) 打开与运行

Python 程序可以用记事本或其他文本编辑工具编写，然后以 `.py` 的扩展名保存到某一文件夹中，再用 IDLE（Python 3）将其打开运行。

① 启动 IDLE，单击 File（文件）菜单，选择 Open（打开）命令，打开保存好的 Python 程序文件（如 `HelloChina.py`）。

② 单击 Run（运行）菜单，选择 Run Module 命令，运行 Python 程序文件。

③ 运行结果在 Python Shell 窗口中显示，具体操作如图 1-9 所示。

2) 新建与保存

启动 IDLE，在 File（文件）菜单中选择 New File（新建文件）命令，进入 Python 编程界面，输入如程序 1-3 所示的程序，单击 File（文件）菜单，选择 Save（保存）命令，然后设置保存路

径、程序文件名称。

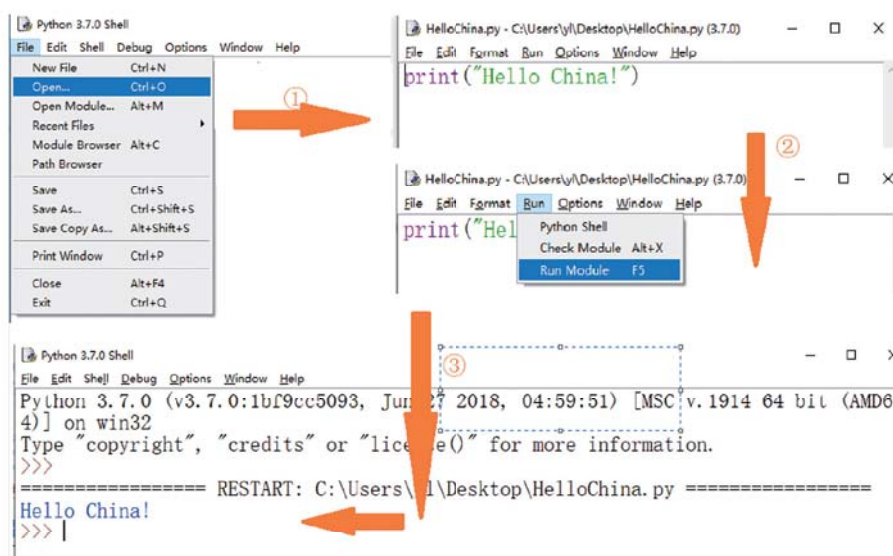


图 1-9 Python 程序文件运行过程

★程序 1-3 祝福祖国

```
name1=input("请输入姓名:")
wish1=input("请输入祝福:")
name2=input("请输入姓名:")
wish2=input("请输入祝福:")
name3=input("请输入姓名:")
wish3=input("请输入祝福:")
print(name1,'祝愿祖国——',wish1)
print(name2,'祝愿祖国——',wish2)
print(name3,'祝愿祖国——',wish3)
```

运行程序 1-3，结果如图 1-10 所示。

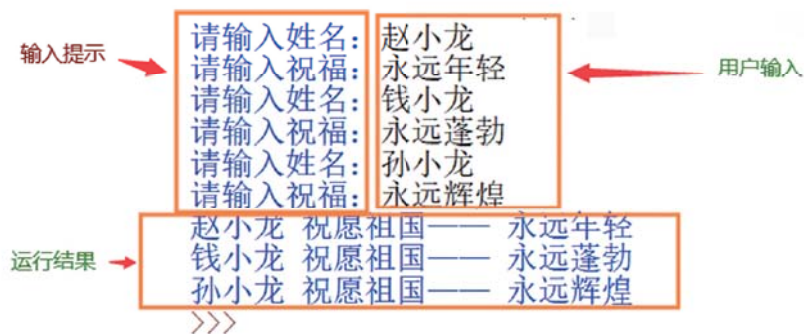


图 1-10 祝福祖国程序及其运行结果

【体验】

新建 Python 程序，在程序文件模式下输入如图 1-11 所示的程序，并保存、运行。

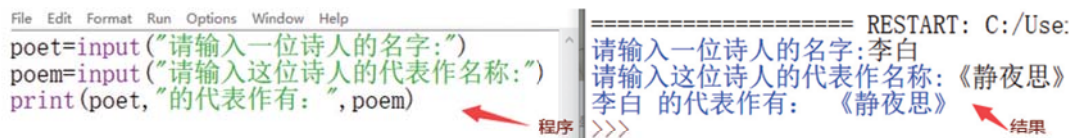


图 1-11 测试程序

3. 在线练习器

在线的 Python 运行环境，无须下载 Python，也不用安装配置，可以直接在线编程并测试。但在线的 Python 不支持输入语句，也不支持外部库，故很多有趣的程序（如海龟画图 etc）不方便在练习器中运行。

在线练习器有两个版本，一是 Python v2.7.13，由 compileonline.com 提供，不支持中文；二是 Python v3，由 runoob.com 提供。在此，我们选择第二个，打开网页，页面左边是写程序代码的地方，单击中部上方的“点击运行”按钮，就可以在页面右边看到程序的结果，如图 1-12 所示。

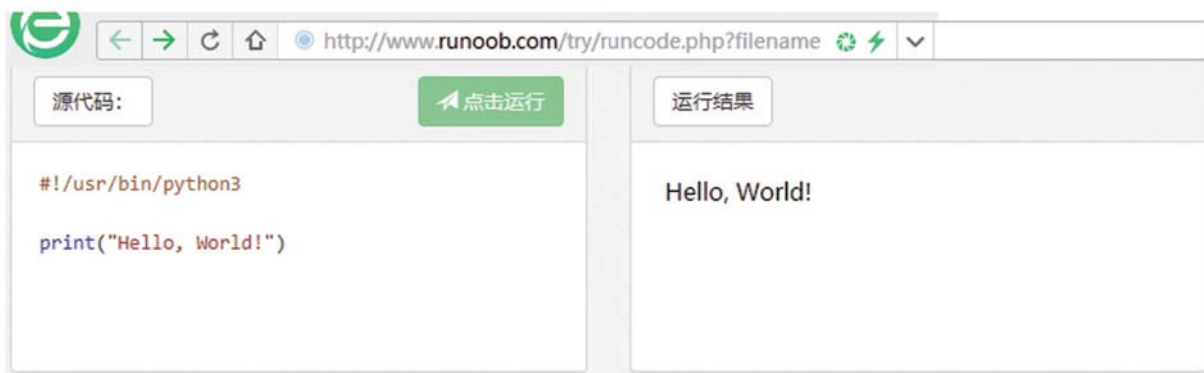


图 1-12 Python v3 在线练习器

【体验】

请通过 Python 在线练习器体验程序 1-1 ~ 程序 1-3 代码的输入和运行。

通过本节的学习，小龙知道了如何使用 Python 平台。

● 项目实施

编程项目的开展离不开编程环境的应用，请利用本节所学知识，积极开展 Python 平台的使用探究，并结合自己的项目主题和方案开展项目实践。



第 2 节 Python 程序设计基础知识

● 小龙的想法

小龙用 Python 编写诗词测试游戏，进行古今对话的探究遇到了困难！他想给 Python “下指令”，却不知道要怎么“说”。他要学会如何把他的“指令”，变成 Python 能够解读并执行的命令，就要先去了解 Python 的“语法规则”，于是，他开始探究什么是常量和变量，Python 的数据类型，以及运算符和表达式等内容。

在正式开始 Python 程序设计之前，我们首先得了解一些 Python 程序设计的基础知识，包括常量和变量、数据类型、运算符与表达式、函数等概念。

一、常量和变量

编程主要是对各种数据（可能是数字、字母、图片、文件等类型）进行不同的处理（输入、输出、查找、转换等）。数据总是以常量或变量两种形式存在于程序中。

1. 常量

常量是指在程序运行过程中不会发生变化的量，通常是固定的字符串或数值。如 "李白"、"HELLO"、3.14159、5 等都是常量。在 Python 3 中，字符串常量应该置于一对英文的单引号或双引号之间，比如 'HELLO'、"HELLO" 都可以。

2. 变量

变量是在程序运行过程中值可以发生变化的量。

变量包含两部分：变量名和变量值。

如：name = 'Lilei' 这一语句中，name 就是一个变量名称，'Lilei' 就是这个变量的值，这个值本身是一个字符串。

Python 中，“=”表示赋值，即将“=”右边的值赋给左边的变量。

又如：x = 9 * (5 + 7) 这一语句中，x 是变量名，它的值是表达式“9 * (5 + 7)”的结果，即 108。

【思考】

在图 1-8 和图 1-11 中有常量和变量吗？哪些是常量，哪些是变量？

变量的命名要遵守 Python 的标识符命名规定。

- (1) 只能由字母（a ~ z、A ~ Z）、数字 0 ~ 9 以及下划线组成。
- (2) 变量名的第一个字母不能是数字。
- (3) 不能用 Python 已经定义的具有特定作用的关键词做变量名，如“def”。
- (4) 区分大小写，如 a1 和 A1 是不同的变量名。

【实践】

在 Python 中输入如图 1-13 所示的程序语句，预估 Python 的输出结果并与实际输出结果进行比较，理解变量的概念。

```
>>> x=9*(5+7)
>>> x=5+7
>>> print(x)
```

图 1-13 实践测试程序

二、数据类型

数值和字符串是最常用的两种数据类型，但是 Python 能直接处理的常用数据类型并不仅这两种。Python 常用的基本数据类型如表 1-2 所示。

表 1-2 Python 常用的基本数据类型

数据类型	类型说明	示 例
Number (数值)	int (整型)：正负整数。在 Python 3 中，无论整数的大小长度是多少，都是 int 整型。 float (浮点型)：包含小数。也可以用科学记数法表示	x=1234567891011121314151617 x 的值是整型 p=3.1415926535898 p 的值是浮点型 k=3.14e3 k 的值是 3140.0
String (字符串)	字符串是以英文单引号或双引号括起来的任意文本。字符串里的字符是有序的，序号(索引号)从 0 开始，-1 是倒数第一个字符的索引	s1='Hello' s2="Hi" s1[0] 即 'Hello' 字符串中的第一个字符 'H'； s1[-1] 即 'Hello' 字符串中的倒数第一个字符 'o'
List (列表)	列表是一种有序集合，列表内的元素用方括号“[]”括起，每个元素用逗号隔开。元素可以是不同类型的，并且可以修改。列表索引也是从 0 开始	L=[] L=[1, 2, 3], 'a', 3.14] L[0] 是 [1, 2, 3] L[2] 是 3.14
Tuple (元组)	元组是不可以修改的列表，元素用圆括号“()”括起	T= ('name', 'age', 'Number') T[0] 是 'name'
Set (集合)	集合是一个无序、不重复的元素集合，同一集合中只能存储不可变的数据类型，比如数值、字符串、元组。创建空集用 set()。给集合赋值用 {}。集合可以用来求“交、差、并、补”等关系	s0 = set() s1 = { 11, 22, 33, 44 } s2 = { 11, 55, 66, 77 } 求交集用 s1 & s2 得到结果 { 11 }
Dictionary (字典)	字典是无序的“键：值”对的集合，用大括号“{ }”括起，每个“键”与一个“值”相对应，每个字典中，“键”是唯一的，使用“键”来访问它对应的“值”	D = {'No1' : 97, 'No2' : 95, 'No3' : 94} print (D['No1']) 则输出 97



【交流】

在程序 1-2 中使用了哪些数据类型？

三、运算符与表达式

设计程序时经常要对参与运算的对象（称为操作数，可以是常量、变量、函数）进行运算或处理，不同的运算要用到不同的运算符。由运算符和操作数连接而成的式子称为表达式。常用的运算符及表达式示例如表 1-3 所示。

表 1-3 运算符与表达式

运算类型及运算符		描 述	示例：x = 5, y = 3	
			表达式	表达式结果
算术运算 (用于数学运算)	+	加	x + y	8
	-	减	x - y	2
	*	乘	x * y	15
	**	幂	x ** y	125
	/	除	x / y	1.6666666666666667
	//	整除(除后向下取整,不是四舍五入)	x // y	1
	%	求余数	x % y	2
关系运算 (关系判断, 结果为 False 或 True, 可对应于整数 1 或 0)	<	小于	x < y	False
	<=	小于等于	x <= y	False
	>	大于	x > y	True
	>=	大于等于	x >= y	True
	==	等于	x == y	False
	!=	不等于	x != y	True
逻辑运算 (结果为 False 或 True, 可对应于整数 1 或 0)	and	“与”。a and b, 若 a 为 False, 表达式结果为 False	x < y and y > 0	False
		a and b, 若 a 为 True, 表达式结果为 b 的值	x > y and y	3
	or	“或”。a or b, 若 a 为 True, 表达式结果为 a 的值	x > y or x	True
		a or b, 若 a 为 False, 表达式结果为 b 的值	x < y or y	3
	not	“非”，若 a 为 True, not a 是 False; 若 a 为 False, not a 是 True	not x < y	True

续表

运算类型 及运算符	描 述	示例: $x = 5, y = 3$	
		表达式	表达式结果
赋值 (改变的是左边的变量值)	=	赋值	$x = y$ x 的值为 3
	+=	加后赋值: $a += b$, 即 $a = a + b$	$x += y$ x 的值为 8
	-=	减后赋值: $a -= b$, 即 $a = a - b$	$x -= y$ x 的值为 2
	*=	乘后赋值: $a *= b$, 即 $a = a * b$	$x *= y$ x 的值为 15
	/=	除后赋值: $a /= b$, 即 $a = a / b$	$x /= y$ x 的值为 1.6666666666666667
	//=	除后舍余并赋值: $a //= b$, 即 $a = a // b$	$x //= y$ x 的值为 1
	%=	求余后赋值: $a %= b$, 即 $a = a \% b$	$x \% = y$ x 的值为 2
	**=	求幂后赋值: $a ** = b$, 即 $a = a ** b$	$x ** = y$ x 的值为 125

当表达式中包含多种运算时, 将根据运算符的优先级按由高到低的顺序进行运算。如果是同级运算则从左至右。如图 1-14 所示, 从上至下为优先级由高到低。

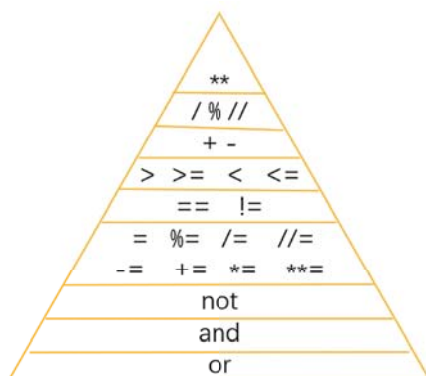


图 1-14 运算符优先级示意图

四、函数

函数是事先组织好、可以重复使用的具有特定功能的代码段, 比如求平方根函数、输入输出函数、类型转换函数等。

在数据的处理中, 输入输出数据是比较常用的操作。

1. 输入函数

`input()` 函数可以接受用户键盘输入的内容, 其一般格式为:

```
<变量>=input("提示信息: ")
```

计算机执行此指令时, 将等待用户输入, 其输入值可存入某一变量中。其中, "提示信息:" 为字符串, 若有多个字符串, 则需要用 "+" 连接。在 Python 3 中, 不论用户输入什么类型的数据,



`input()` 函数的返回结果都是字符串，需要将其转换为相应的类型再处理。

【交流】

```
poet=input("请输入《静夜思》的作者：")
```

上面这行程序代码中，变量名、提示信息分别是什么？变量值是什么？

你的项目主题中，需要进行数据“输入”吗？你要如何写输入语句？

2. 输出函数

用 `print()` 函数可以将数据输出到 Python Shell 上进行显示，其一般格式为：

```
print( object , sep = ' ', end = '\n')
```

(1) `object` 是输出对象，可以是一个或者多个对象（常量、变量、表达式等），对象之间用逗号分隔。

(2) `sep` 参数不是必需的，它的作用是控制每个输出对象之间的分隔符，如果不设置，默认是一个空格。

(3) `end` 后面的字符表示本句 `print()` 函数最后一个输出对象后面要显示的字符串。如果缺省，默认是一个“`\n`”换行符。如果设为其他字符，则输出当前行所有内容后，在末尾加该符号，不换行，继续等待下一个 `print()` 的输出对象。

例如，`print(1,2,3,sep='#')` 的输出结果为“1#2#3”。

3. 数据类型转换函数

有时候，我们需要将一种数据类型转换为其他数据类型，表 1-4 为三种常用的数据类型转换函数。

表 1-4 常用数据类型转换函数

函数名	作用	示例	结果
<code>float()</code>	将括号里的值转换为浮点类型	<code>float(2)</code>	2.0
<code>int()</code>	将括号里的值转换为整数类型（舍小数）	<code>int(3.56)</code>	3
<code>str()</code>	将括号里的值转换为字符串类型	<code>str(3.14)+str(0.5)</code>	'3.140.5'

【实验】

在交互会话模式下输入表 1-5 左列的内容，记录输出，并分析和讨论右列的问题。

表 1-5 交互模式输入输出示例表

输入	输出	分析问题
<code>x=input("请输入今年的年份：")</code>		<code>input()</code> 函数的作用
<code>print(x-1949)</code>		变量 <code>x</code> 的值和数据类型
<code>y=int(input("请输入今年的年份："))</code>		<code>int()</code> 的作用是什么

续表

输 入	输 出	分析问题
<code>print(y-1949)</code>		变量 y 的值和数据类型
<code>print(' 今年是新中国成立 ',y-1949,' 年 ')</code>		<code>print()</code> 如何显示多项内容

通过本节的学习，小龙初步掌握了编写 Python 程序的一些基础知识。他要编写的诗词测试程序可能暂时还用不到所有的数据类型和函数，但是他已经在思考进行游戏测试时，会有哪些输入和输出，它们应该是什么数据类型，它们是变量还是常量，要对它们做什么样的分析和处理。

● 项目实施

请根据小组选定的项目方案，结合本节所学知识，分析项目中所要用到的数据类型，并进行数据的输入、输出测试，进一步讨论完善该项目方案中的各项探究活动，并进行探究实践。

第 3 节 顺序结构程序设计

● 小龙的想法

小龙发现编写程序就像导演拍戏一样，导演给出一个指令，演员就按照指令进行表演。通过前面的探究，他感觉不同类型的数据就像不同的演员，每个表达式就像一幕幕场景，函数就像一个个动作指令。你方唱罢我登场，按顺序执行的程序最简单。他决定就用输入和输出语句来“玩”诗词游戏。

程序结构包括顺序结构、选择结构、循环结构三种，本节开始学习 Python 的顺序结构程序设计。

一、Python 的顺序结构

顺序结构是最简单的程序结构，由若干个依次执行的处理步骤组成。如图 1-15 所示，语句 A 和语句 B 是依次执行的，只有在执行完语句 A 后，才能接着执行语句 B。

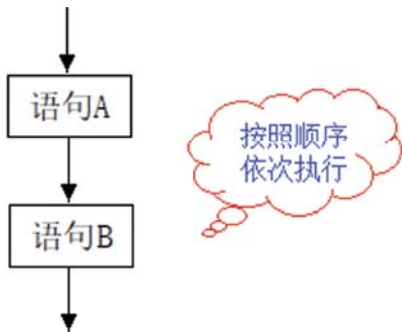


图 1-15 程序的顺序结构



二、顺序结构的应用

在 Python 的顺序程序设计中，最典型的应用就是包含输入、赋值和输出指令的程序。

★程序 1-4

```
poet=input("请出题——输入一位诗人的名字：")
poem=input("请回答——输入这位诗人的一首代表诗作：")
print(poet,'的代表作:',poem)
```

程序 1-4 的目标是输入一位诗人的名字，然后再输入该诗人的代表作，最后输出显示。

★程序 1-5 根据半径计算圆的面积

```
r=float(input("请输入圆的半径："))    #float 浮点数
s=3.1415926535*r**2                    # 将圆周率 Pi 约记为 3.1415926535
print("圆的面积=",s)
```

程序 1-5 的目标是输入圆的半径，计算并显示圆的面积。

思路分析如下。

- (1) 用户输入圆的半径，并存入半径变量 `r` 中。
- (2) 用圆面积公式计算圆的面积。（注意 “`**`” 运算符的作用）
- (3) 将圆的面积输出到屏幕。

【实验】

程序 1-5 中，用了转换函数 `float()`，如果不用这个转换函数，程序能计算出正确的结果吗？请进行尝试并观察结果。

程序 1-5 中 “`#`” 的作用是表示注释，注释的作用是解释语句或程序功能，仅供阅读，不参与运行。

Python 中的注释分为单行注释和多行注释，如图 1-16 所示。

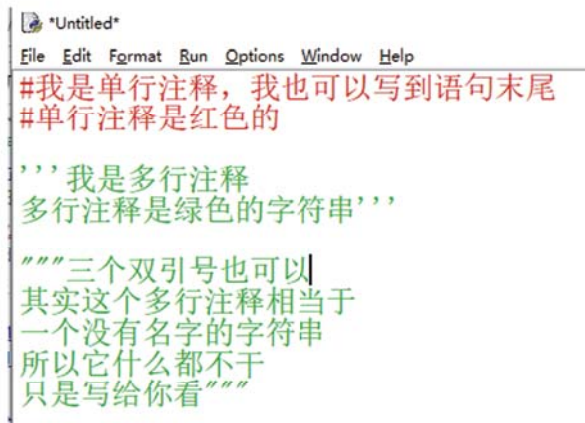


图 1-16 注释使用示例

单行注释用“#”开头，后面加一行注释内容，“#”后面的注释是会被运行的。
多行注释用一对“三引号”表示，其中可以包含多行字符串，它们同样不会运行。

★程序 1-6 变量值交换

```
a=input("请输入数字 1:")
b=input("请输入数字 2:")
print("交换前:",a,b)
c=a
a=b
b=c
print("交换后:",a,b)
```

程序 1-6 的目标是输入两个数，然后交换它们，再输出交换后的结果。

思路分析如下。

让用户分别输入两个数，并存入 a、b 两个变量中；引入第三个变量 c 来交换两个数；显示交换前、后的数字。

【分析】

程序 1-6 的变量 a,b,c 是什么数据类型？如果要让 a,b 进行数学的加减运算，程序 1-6 可以实现吗？

【阅读】

(1) 读取用户输入的两个数，在 Python 中可优雅地表示为：

```
a,b=input("请输入数字 1:"),input("请输入数字 2:")
```

(2) a、b 数据的交换，也可优雅地表示为：

```
a,b=b,a
```

【实践】

- (1) 编写计算梯形面积的 Python 程序，并测试运行。
- (2) 按顺序输出清朝的所有皇帝名称。
- (3) 按顺序输出一首歌的歌词。

通过本节的学习，小龙感觉写顺序结构的 Python 程序很简单，他“清点”了一下自己的技能：输入、输出、表达式和赋值等。用这些技能他可以实现让诗词按程序语句的要求输入或者输出。按顺序，一切都那么自然。

● 项目实施

请根据小组选定的项目方案，结合本节所学知识，利用顺序结构进行方案的设计和实现，进一步讨论完善该项目方案中的各项探究活动，并进行探究实践。