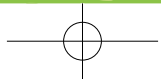
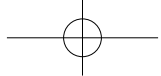


第 3 单元

奇妙的数字





第3单元 奇妙的数字



“小萌，你还记得天宫课堂吗？”

“当然记得，我亲眼看到了王亚平阿姨演示太空中失重环境下的好多现象，太神奇了！”



“对，之所以能天宫授课，都是因为航天技术的发展。航天技术是计算机在科学计算方面的重要应用，因为计算机诞生之初就是为了科学计算。你知道科学计算的基本处理对象是什么吗？”

“嗯，计算嘛，肯定是处理数字啰！”



“回答正确，现在就让我们一起来认识编程中奇妙的数字吧！”

数字是数学运算和推理表示的基础，数字类型是 Python 的基础数据类型之一。这里，我们先来认识 Python 中整数、浮点数、真假值和空值这几种基本数字类型的表示方式，并学会对它们进行基本的算术运算，迈出科学计算的第一步。



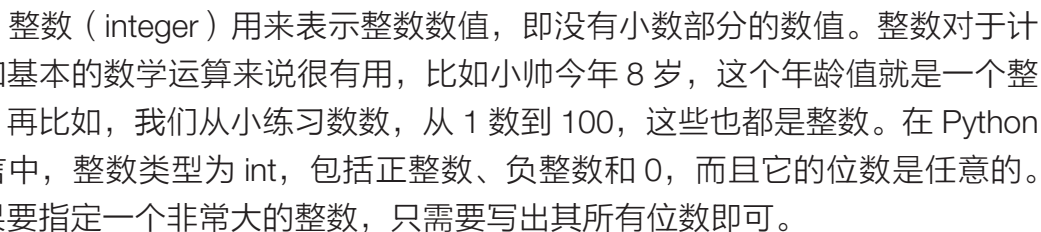
3.1 整数



“小帅，你几岁了？”

“我8岁。”

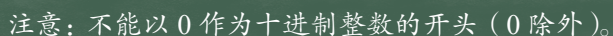




在 Python 中，整数都是 `int`（integer）对象的实例，其字面值以十进制（decimal）为主，特定情况下可以使用二进制（binary）、八进制（octal）或十六进制（hexadecimal）表示。

十进制整数

十进制整数的表现形式大家都很熟悉。例如，下面的数值和算式都是有效的。

[illegible]

二进制整数

由 0 和 1 两个数组成，进位规则是“逢二进一”，并且以 0b/0B 开头，例如 0b101（转换为十进制整数为 5）或 0B1111（转换为十进制整数为 15）。





```
>>>0b101
5
>>>0b1111
15
```



八进制整数

由 0 ~ 7 组成,进位规则为“逢八进一”,并且以 0o/0O 开头,例如 0o20(转换为十进制整数为 16)或 0O37(转换为十进制整数为 31)。

```
>>>0o20
16
>>>0O37
31
```



十六进制整数

由 0 ~ 9、A ~ F (a ~ f) 组成,进位规则为“逢十六进一”,并且以 0x/0X 开头,例如 0xf(转换为十进制整数为 15)或 0X3a(转换为十进制整数为 58)。

```
>>>0xf
15
>>>0X3a
58
```



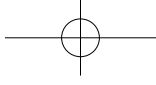
十进制整数

可以通过内置函数与其他进制进行转换,相关函数如表 3-1 所示。

表 3-1 进制转换内置函数

内 置 函 数	说 明
<code>bin(int)</code>	将十进制数值转换成二进制表示形式,转换的结果以 0b 为前缀





续表

内 置 函 数	说 明
<code>oct(int)</code>	将十进制数值转换成八进制表示形式，转换的结果以 <code>0o</code> 为前缀
<code>hex(int)</code>	将十进制数值转换成十六进制表示形式，转换的结果以 <code>0x</code> 为前缀
<code>int(s,base)</code>	将字符串 <code>s</code> 依据 <code>base</code> 参数提供的进制数转换成十进制数



“小萌，你知道著名的‘五四运动’发生在哪一年吗？”

“我知道，是1919年。”



“太棒了，那我们一起用内置函数来看看1919的各种进制对应的数值吧。”



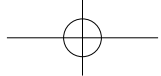
【问题 3-1】 以下代码的输出结果是什么？

```
print(" 类型: ",type(1919))
print(" 二进制: ",bin(1919))
print(" 八进制: ",oct(1919))
print(" 十六进制: ",hex(1919))
```

【问题 3-2】 为什么下面所列举的整数是不合法的？

```
0456
0b123
0o789
0X5G
```





练一练

【问题 3-3】关于 Python 中整数的表示形式，错误的是（ ）。

- A. 1919
- B. 0B123
- C. 0o456
- D. 0x789

3.2 浮点数



“你们听说过祖率吗？”

“老师，我知道！我知道祖率就是圆周率 π ，3.1415926。”



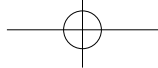
“对，祖冲之对圆周率数值的精确推算值，对于中国乃至世界是一个重大贡献，所以后人用他的名字命名圆周率，简称‘祖率’。”

小知识：南北朝时，祖冲之算出的圆周率的近似值在 3.1415926 和 3.1415927 之间，他的圆周率精确值在当时世界遥遥领先，直到 1000 年后阿拉伯数学家阿尔卡西才超过他。



像 3.1415926 这样由整数部分和小数部分组成的数就是浮点数。在 Python 语言中，浮点数类型为 float，例如 1.414、0.5、-1.732 等。

浮点数可以使用 E 记法（也叫科学计数法）表示，用于表示特别大和特别小的数。E 的意思是指数，也可以写作 e，其指底数为 10，E (e) 后边的数字就是 10 的多少次幂。例如，地球到火星的距离平均大约为 2.25 亿千米，等于



2.25×100000000 , 也就是 2.25×10^8 , E 记法写作 $2.25e8$ 。如果给 Python 提供一些极端的数据, 那么它会自动采用 E 记法来表示, 例如, 0.0000000000789 会被记作 $7.89e-11$ 。

```
>>>2.25e8
225000000.0
>>>0.0000000000789
7.89e-11
```



注意: (1) 浮点数的整数或小数部分为 0 时, 书写时可以省略 0, 但不能同时省略 (1. 和 .1 都是合理的浮点数常量)。

(2) 在 E 记法中, 要求 e (或 E) 之前必须有数, 之后的指数必须为整数。



【问题 3-4】 下面所列举的浮点数的 E 记法为什么是不合法的?

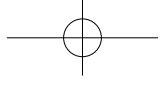
```
1.56E2.3
e5
3.5e-4.5
```



【问题 3-5】 关于 Python 中浮点数的表示, 错误的是 ()。

- A. 8.
- B. -.45
- C. 2.7E3.4
- D. .67e-2





3.3 真值和假值



“小萌,小帅,你们看过中央电视台财经频道的‘是真的吗’这个电视节目吗?”

“我看过,那是一个互动求证节目,每周六晚上 18:30 播出,特别有意思。”



“对,随着微博、博客、社交网站、即时通信、视频分享等传播渠道的发展,信息传播便捷自由,对于公众而言,面对各类传言,真假莫辨,不堪其扰。这个节目携手电视观众与广大网友,通过各大新媒体共同互动求真,对网络流言进行专业验证与权威实验,探求真相。建议你们多看看,培养求真务实的科学精神。”



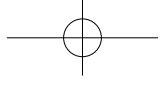
一个句子所描述的东西有对有错,而关于一个句子所描述的东西的对错状况称为真值。例如,“太阳会发光”的说法是对的,其值为真。而“太阳不会发光”的说法是错的,其值为假。

Python 提供了 bool 类型来表示真(对)或假(错)。比如,非零数字在程序世界里视为真(对),Python 使用 True 来代表;而数字零(0 或 0.0)在程序世界里视为假(错),Python 使用 False 来代表。

在 Python 中,所有对象都可以进行真值测试。如果不确定一个对象的真假,可以用内置函数 bool() 来判断。

```
>>>bool(0)
False
>>>bool(0.0)
False
```





```
>>>bool(3.14)
True
>>>bool(1949)
True
>>>bool(0b101)
True
```

值得一提的是，布尔类型可以当作整数来对待，即 True 相当于整数值 1，False 相当于整数值 0。因此，下面这些运算都是可以的：

```
>>>False+1
1
>>>True+1
2
```



注意：True 和 False 是 Python 中的关键字。当作为 Python 代码输入时，一定要注意字母的大小写，否则解释器会报错。

在 Python 中，所有的对象都可以进行真假值的测试，包括字符串、元组、列表、字典、对象等。由于目前尚未学习，因此这里不做过多讲述，后续遇到时会做详细的介绍。

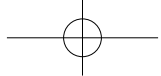
3.4 空 值

在 Python 中，有一个特殊的常量 None（N 必须大写）。和 False 不同，它不表示 0，也不表示空字符串，而表示没有值，也就是空值。

None 有自己的数据类型，可以在 IDLE 中使用内置函数 type() 查看它的类型，执行代码如下：

```
>>>type(None)
<class 'NoneType'>
```





可以看到，它属于 `NoneType` 类型。

需要注意的是，`None` 是 `NoneType` 数据类型的唯一值，也就是说，我们不能再创建其他 `NoneType` 类型的变量，但是可以将 `None` 赋值给任何变量。`None` 常用于 `assert`、判断以及函数无返回值的情况。

3.5 认识表达式

程序语言最大的作用就是将数据经过处理、运算后，转换成有用的信息供我们使用。Python 程序语言有不同种类的运算符，可以和操作数一起组成表达式，然后进行运算。表达式由操作数（operand）与运算符（operator）组成，具体说明如下。

操作数：包括变量、数值和字符。

运算符：包括算术运算符、赋值运算符、逻辑运算符和比较运算符等。

在 Python 中，根据参与运算的操作数的个数将运算符分为单目运算符和双目运算符，本节先介绍算术运算符及算术表达式。

算术运算符是处理类似四则运算的符号，在数字的处理中用得最多。常用的算术运算符如表 3-2 所示。

表 3-2 常用的算术运算符

运算符	含 义	实 例	结 果
+	加	12.5+15	27.5
-	减	4.56-1.36	3.20
*	乘	5*3.6	18.0
/	除	7/2	3.5
%	求余，即返回除法的余数	7%2	1
//	整除，即返回商的整数部分	7//2	3
**	幂，即返回 x 的 y 次方	2**3	8，即 2 ³



注意：使用除法（/ 或 //）运算符和求余运算符时，除数不能为 0，否则将会出现异常，如图 3-1 所示。



```
IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MS
C v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more inf
ormation.
>>> 5//0
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    5//0
ZeroDivisionError: integer division or modulo by zero
>>> 5/0
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    5/0
ZeroDivisionError: division by zero
>>> 5%0
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    5%0
ZeroDivisionError: integer division or modulo by zero
>>>
```

图 3-1 除数为 0 时出现的错误提示

由算术运算符将运算对象连接而成的表达式就是算术表达式。Python 提供的算术运算符，其运算法则与数学中的运算法则相同：先乘除后加减，有括号者优先。



“在进行算术运算时，若参与运算的操作数类型不同，则会发生隐式类型的转换。”

隐式类型转换方式如表 3-3 所示。

表 3-3 隐式类型转换方式

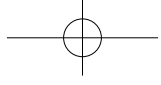
操作数 1 的类型	操作数 2 的类型	隐式转换后的类型
布尔	整数	整数
布尔、整数	浮点	浮点



“要想实现正确的运算，先要能够写出正确的表达式。小萌，你知道代数式 $a \left(1 + \frac{a-b}{100} \right) n$ 在程序代码里应该写成什么样的表达式吗？”

“老师，又是分式又是上标的，没法输入呀……”





第3单元 奇妙的数字



“其实很简单，记住：乘号不能省略，用运算符和括号将式子的运算体现在一行上即可。上面这个代数式应该写作表达式： $a * (1 + (a - b) / 100) ** n$ 。”

“我明白了，就像这个式子里分式用 /，次幂用 **。”



【问题 3-6】代数式 $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$ 对应的表达式应该是什么样的？

【问题 3-7】下面表达式的运算结果各是什么？

```
348/25
348//25
358%25
81**0.5
True+2.5
```



3.6 算术复合赋值运算符



在实际应用中，赋值运算符（=）可以与算术运算、位运算中的二元运算符构成复合赋值运算符，使得程序更加精练，提高效率。当 Python 解释器执行到复合赋值运算符时，先将左边变量与右边表达式做对应的二元运算，然后再将运算结果赋值给左边的变量。这里先介绍算术复合赋值运算符，其形式及运算规则如表 3-4 所示。





表 3-4 算术复合赋值运算符

运 算 符	含 义	举 例	展 开 形 式
+=	加赋值	$x+=y$	$x=x+(y)$
-=	减赋值	$x-=y$	$x=x-(y)$
=	乘赋值	$x=y$	$x=x*(y)$
/=	除赋值	$x/=y$	$x=x/(y)$
%=	取余数赋值	$x\%=y$	$x=x\%(y)$
//=	取整除赋值	$x//=y$	$x=x//(y)$
=	幂赋值	$x=y$	$x=x**(y)$



注意：在复合赋值运算符中，默认将右边部分视作一个整体。



【问题 3-8】 以下程序段执行完后，x 的值是什么？

```
x, y=12, 3
x += y
x //= y+2
x **= y-1
```

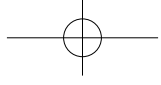
3.7 牛刀小试



“小帅，你知道‘好好学习，天天向上’的典故是出自哪里的吗？”

“好像是毛主席给一个小学生写的题词。”





第3单元 奇妙的数字



“对，这句话现在已经成为鼓励大家好好读书的代表了。持之以恒就能取得好成绩，反之则差距很大，下面我们一起来编程计算一下。”

【例 3-1】天天向上的力量。

一年 365 天，以第 1 天的能力值为基数，记为 1.0，当好好学习时，能力值相比前一天提高 1%，当没有学习时，由于遗忘等原因，能力值相比前一天下降 1%。每天努力和每天放任，一年下来的能力值相差多少呢？

程序代码如下：

```
dayup=(1+0.01)**365
daydown=(1-0.01)**365
print(" 向上 :",dayup," "," 向下 :",daydown)
```

程序的运行结果如下：

向上：37.78343433288728 ， 向下：0.025517964452291125



“哇，每天努力，一年下来能力值提高了 37 倍啊！”

“天哪，每天放任的话，一年下来能力值基本归零了！”



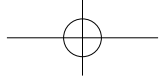
“所以，学习和做事都贵在坚持啊！”

“嗯嗯，我们一定会好好努力的。”



“本单元重点介绍了 Python 中的基本数字类型、运算符和表达式。要求同学们能准确使用各种类型的数字，并能熟练地对其进行运算，为后续的代码编写打下良好基础。”

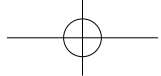




习 题 3

1. 小明的身高是 153.4cm, Python 中不能正确表达该身高的数值是 ()。
A. 153.40 B. 1.534E2 C. 1534 / 10 D. 153.4cm
2. x 是一个两位的整数变量, 将其十位数与个位数交换位置的语句是 ()。
A. $(x/10)\%10 + x//10$ B. $(x/10)\%10 + x\%10$
C. $(x\%10)*10 + x\%10$ D. $(x\%10)*10 + x//10$
3. 关于整数类型的 4 种进制表示, 描述正确的是 ()。
A. 二进制、四进制、八进制、十进制
B. 二进制、四进制、十进制、十六进制
C. 二进制、四进制、八进制、十六进制
D. 二进制、八进制、十进制、十六进制
4. 以下不是 Python 语言的整数类型的选项是 ()。
A. 0B1010 B. 88 C. 0x9a D. 0E99
5. hex(255) 的执行结果是 ()。
A. '0xff' B. '-0xff' C. 0xff.0 D. 0xff
6. Python 语言 % 运算符的含义是 ()。
A. x 与 y 之商 B. x 与 y 的整数商
C. x 与 y 之商的余数 D. x 的 y 次幂
7. 以下 Python 浮点数类型错误的是 ()。
A. 0.0 B. 96e4 C. -0x89 D. 9.6E5





第3单元 奇妙的数字



8. $100 // 3$ 的执行结果是 ()。

A. 3

B. 33

C. 0.3333333333333336

D. 33.333333333333336

9. $100 / .3$ 的运算结果是 ()。

A. 3

B. 33

C. 33.333333333333336

D. 333.33333333333337

10. $100 ** (1/2)$ 的运算结果是 ()。

A. 50.0

B. 10.0

C. 50

D. 10

11. 编写程序实现如下功能：

小明的考试成绩为语文 78 分、数学 85 分、英语 92 分。从键盘上输入这 3 科的分数，并计算它们的总分和平均分。

