

语言元素

指令和程序

计算机的硬件系统通常由五大部件构成，包括：运算器、控制器、存储器、输入设备和输出设备。其中，运算器和控制器放在一起就是我们通常所说的中央处理器，它的功能是执行各种运算和控制指令以及处理计算机软件中的数据。我们通常所说的程序实际上就是指令的集合，我们程序就是将一系列的指令按照某种方式组织到一起，然后通过这些指令去控制计算机做我们想让它做的事情。今天我们大多数时候使用的计算机，虽然它们的元器件做工越来越精密，处理能力越来越强大，但究其本质来说仍然属于“冯·诺依曼结构”的计算机。“冯·诺依曼结构”有两个关键点，一是指出要将存储设备与中央处理器分开，二是提出了将数据以二进制方式编码。二进制是一种“逢二进一”的计数法，跟我们人类使用的“逢十进一”的计数法没有实质性的区别，人类因为有十根手指所以使用了十进制（因为在数数时十根手指用完之后就只能进位了，当然凡事都有例外，玛雅人可能是因为长年光着脚的原因把脚趾头也算上了，于是他们使用了二十进制的计数法，在这种计数法的指导下玛雅人的历法就与我们平常使用的历法不一样，而按照玛雅人的历法，2012年是上一个所谓的“太阳纪”的最后一年，而2013年则是新的“太阳纪”的开始，后来这件事情被以讹传讹的方式误传为“2012年是玛雅人预言的世界末日”这种荒诞的说法，今天我们可以大胆的猜测，玛雅文明之所以发展缓慢估计也与使用了二十进制有关）。对于计算机来说，二进制在物理器件上来说是更容易实现的（高电压表示1，低电压表示0），于是在“冯·诺依曼结构”的计算机都使用了二进制。虽然我们并不需要每个程序员都能够使用二进制的思维方式来工作，但是了解二进制以及它与我们生活中的十进制之间的转换关系，以及二进制与八进制和十六进制的转换关系还是有必要的。如果你对这一点不熟悉，可以自行使用[维基百科](#)或者[百度百科](#)科普一下。

提示：近期关于量子计算机的研究已经被推倒了风口浪尖，量子计算机基于量子力学进行运算，使用量子瞬移的方式来传递信息。2018年6月，Intel宣布开发出新款量子芯片并通过了在接近绝对零度环境下的测试；2019年1月，IBM向全世界发布了首款商业化量子计算机。

变量和类型

在程序设计中，变量是一种存储数据的载体。计算机中的变量是实际存在的数据或者说是存储器中存储数据的一块内存空间，变量的值可以被读取和修改，这是所有计算和控制的基础。计算机能处理的数据有很多种类型，除了数值之外还可以处理文本、图形、音频、视频等各种各样的数据，那么不同的数据就需要定义不同的存储类型。Python中的数据类型很多，而且也允许我们自定义新的数据类型（这一点在后面会讲到），我们先介绍几种常用的数据类型。

- 整型：Python中可以处理任意大小的整数（Python 2.x中有int和long两种类型的整数，但这种区分对Python来说意义不大，因此在Python 3.x中整数只有int这一种了），而且支持二进制（如0b100，换算成十进制是4）、八进制（如0o100，换算成十进制是64）、十进制（100）和十六进制（0x100，换算成十进制是256）的表示法。
- 浮点型：浮点数也就是小数，之所以称为浮点数，是因为按照科学记数法表示时，一个浮点数的小数点位置是可变的，浮点数除了数学写法（如123.456）之外还支持科学计数法（如1.23456e2）。
- 字符串型：字符串是以单引号或双引号括起来的任意文本，比如'hello'和"hello"，字符串还有原始字符串表示法、字节字符串表示法、Unicode字符串表示法，而且可以书写成多行的形式（用三个单引号或三个双引号开头，三个单引号或三个双引号结尾）。
- 布尔型：布尔值只有True、False两种值，要么是True，要么是False，在Python中，可以直接用True、False表示布尔值（请注意大小写），也可以通过布尔运算计算出来（例如3 < 5会产生布尔值True，而2 == 1会产生布尔值False）。
- 复数型：形如3+5j，跟数学上的复数表示一样，唯一不同的是虚部的i换成了j。实际上，这个类型并不能算作常用类型，大家了解下就可以了。

变量命名

对于每个变量我们需要给它取一个名字，就如同我们每个人都有属于自己的响亮的名字一样。在Python中，变量命名需要遵循以下这些必须遵守硬性规则和强烈建议遵守的非硬性规则。

- 硬性规则：
 - 变量名由字母（广义的Unicode字符，不包括特殊字符）、数字和下划线构成，数字不能开头。
 - 大小写敏感（大写的a和小写的A是两个不同的变量）。
 - 不要跟关键字（有特殊含义的单词，后面会讲到）和系统保留字（如函数、模块等的名字）冲突。
- PEP 8要求：
 - 用小写字母拼写，多个单词用下划线连接。
 - 受保护的实例属性用单个下划线开头（后面会讲到）。
 - 私有的实例属性用两个下划线开头（后面会讲到）。

当然，作为一个专业的程序员，给变量（事实上应该是所有的标识符）命名时做到见名知意也是非常重要的。

变量的使用

下面通过几个例子来说明变量的类型和变量使用。

```
"""
使用变量保存数据并进行算术运算

Version: 0.1
Author: 骆昊
"""

a = 321
b = 123
print(a + b)
print(a - b)
print(a * b)
print(a / b)
print(a // b)
print(a % b)
print(a ** b)
```

在Python中可以使用 `type` 函数对变量的类型进行检查。程序设计中函数的概念跟数学上函数的概念是一致的，数学上的函数相信大家并不陌生，它包括了函数名、自变量和因变量。如果暂时不理解这个概念也不要紧，我们会在后续的章节中专门讲解函数的定义和使用。

```
"""
使用type()检查变量的类型

Version: 0.1
Author: 骆昊
Date: 2018-02-27
"""

a = 100
b = 12.345
c = 1 + 5j
d = 'hello, world'
e = True
```

```
print(type(a)) # <class 'int'>
print(type(b)) # <class 'float'>
print(type(c)) # <class 'complex'>
print(type(d)) # <class 'str'>
print(type(e)) # <class 'bool'>
```

可以使用Python中内置的函数对变量类型进行转换。

- `int()`：将一个数值或字符串转换成整数，可以指定进制。
- `float()`：将一个字符串转换成浮点数。
- `str()`：将指定的对象转换成字符串形式，可以指定编码。
- `chr()`：将整数转换成该编码对应的字符串（一个字符）。
- `ord()`：将字符串（一个字符）转换成对应的编码（整数）。

下面的代码通过键盘输入两个整数来实现对两个整数的算术运算。

```
"""
使用input()函数获取键盘输入(字符串)
使用int()函数将输入的字符串转换成整数
使用print()函数输出带占位符的字符串

Version: 0.1
Author: 骆昊
"""

a = int(input('a = '))
b = int(input('b = '))
print('%d + %d = %d' % (a, b, a + b))
print('%d - %d = %d' % (a, b, a - b))
print('%d * %d = %d' % (a, b, a * b))
print('%d / %d = %f' % (a, b, a / b))
print('%d // %d = %d' % (a, b, a // b))
print('%d %% %d = %d' % (a, b, a % b))
print('%d ** %d = %d' % (a, b, a ** b))
```

说明：上面的print函数中输出的字符串使用了占位符语法，其中`%d`是整数的占位符，`%f`是小数的占位符，`%%`表示百分号（因为百分号代表了占位符，所以带占位符的字符串中要表示百分号必须写成`%%`），字符串之后的`%`后面跟的变量值会替换掉占位符然后输出到终端中，运行上面的程序，看看程序执行结果就明白啦。

运算符

Python支持多种运算符，下表大致按照优先级从高到低的顺序列出了所有的运算符，运算符的优先级指的是多个运算符同时出现时，先做什么运算然后再做什么运算。除了我们之前已经用过的赋值运算符和算术运算符，我们稍后会陆续讲到其他运算符的使用。

运算符	描述
<code>[]</code> <code>[:]</code>	下标, 切片
<code>**</code>	指数
<code>~</code> <code>+</code> <code>-</code>	按位取反, 正负号
<code>*</code> <code>/</code> <code>%</code> <code>//</code>	乘, 除, 模, 整除
<code>+</code> <code>-</code>	加, 减
<code>>></code> <code><<</code>	右移, 左移
<code>&</code>	按位与
<code>^</code> <code> </code>	按位异或, 按位或
<code><=</code> <code><</code> <code>></code> <code>>=</code>	小于等于, 小于, 大于, 大于等于
<code>==</code> <code>!=</code>	等于, 不等于
<code>is</code> <code>is not</code>	身份运算符
<code>in</code> <code>not in</code>	成员运算符
<code>not</code> <code>or</code> <code>and</code>	逻辑运算符
<code>=</code> <code>+=</code> <code>-=</code> <code>*=</code> <code>/=</code> <code>%=</code> <code>//=</code> <code>**=</code> <code>&=</code> <code> =</code> <code>^=</code> <code>>>=</code> <code><<=</code>	(复合) 赋值运算符

说明： 在实际开发中，如果搞不清楚运算符的优先级，可以使用括号来确保运算的执行顺序。

下面的例子演示了赋值运算符和复合赋值运算符的使用。

```
"""
赋值运算符和复合赋值运算符

Version: 0.1
Author: 骆昊
"""

a = 10
b = 3
a += b # 相当于: a = a + b
a *= a + 2 # 相当于: a = a * (a + 2)
print(a) # 想想这里会输出什么
```

下面的例子演示了比较运算符（关系运算符）、逻辑运算符和身份运算符的使用。

```
"""
比较、逻辑和算身份运算符的使用

Version: 0.1
Author: 骆昊
"""

flag0 = 1 == 1
flag1 = 3 > 2
```

```
flag2 = 2 < 1
flag3 = flag1 and flag2
flag4 = flag1 or flag2
flag5 = not (1 != 2)
print('flag0 =', flag0) # flag0 = True
print('flag1 =', flag1) # flag1 = True
print('flag2 =', flag2) # flag2 = False
print('flag3 =', flag3) # flag3 = False
print('flag4 =', flag4) # flag4 = True
print('flag5 =', flag5) # flag5 = False
print(flag1 is True) # True
print(flag2 is not False) # False
```

练习

练习1：华氏温度转换为摄氏温度。

提示：华氏温度到摄氏温度的转换公式为： $C = (F - 32) \div 1.8$ 。

参考答案：

```
"""
将华氏温度转换为摄氏温度

Version: 0.1
Author: 骆昊

f = float(input('请输入华氏温度: '))
c = (f - 32) / 1.8
print('%.1f华氏度 = %.1f摄氏度' % (f, c))
```

练习2：输入圆的半径计算计算周长和面积。

参考答案：

```
"""
输入半径计算圆的周长和面积

Version: 0.1
Author: 骆昊

import math

radius = float(input('请输入圆的半径: '))
perimeter = 2 * math.pi * radius
area = math.pi * radius * radius
print('周长: %.2f' % perimeter)
print('面积: %.2f' % area)
```

练习3：输入年份判断是不是闰年。

参考答案：

```
"""
```

```
输入年份 如果是闰年输出True 否则输出False
```

```
Version: 0.1
```

```
Author: 骆昊
```

```
"""
```

```
year = int(input('请输入年份: '))
```

```
# 如果代码太长写成一行不便于阅读 可以使用\对代码进行折行
```

```
is_leap = (year % 4 == 0 and year % 100 != 0) or \  
          year % 400 == 0
```

```
print(is_leap)
```