

# Day02 - 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。今天我们使用的计算机虽然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。

## 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中类型，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`。

## 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)
```

```
"""
使用input函数输入
使用int()进行类型转换
用占位符格式化输出的字符串

```

```
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))
```

```
"""
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))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
```

在对变量类型进行转换时可以使用Python的内置函数（准确的说下面列出的并不是真正意义上的函数，而是后面我们要讲到的创建对象的构造方法）。

- 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）。

## 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，我们会陆续使用到它们。

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>&gt;&gt;</code> <code>&lt;&lt;</code>	右移, 左移
<code>&amp;</code>	按位与
<code>^</code> <code> </code>	按位异或, 按位或
<code>&lt;=</code> <code>&lt;</code> <code>&gt;</code> <code>&gt;=</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>&amp;=</code> <code> =</code> <code>^=</code> <code>&gt;&gt;=</code> <code>&lt;&lt;=</code>	(复合) 赋值运算符

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

下面的例子演示了运算符的使用。

```
"""
运算符的使用

Version: 0.1
Author: 骆昊
"""

a = 5
b = 10
c = 3
d = 4
e = 5
a += b
a -= c
a *= d
a /= e
print("a = ", a)

flag1 = 3 > 2
flag2 = 2 < 1
```

```
flag3 = flag1 and flag2
flag4 = flag1 or flag2
flag5 = not flag1
print("flag1 = ", flag1)
print("flag2 = ", flag2)
print("flag3 = ", flag3)
print("flag4 = ", flag4)
print("flag5 = ", flag5)
print(flag1 is True)
print(flag2 is not False)
```

## 练习

### 练习1：华氏温度转摄氏温度。

```
"""
将华氏温度转换为摄氏温度
F = 1.8C + 32

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。

```
"""
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。

```
"""
```

```
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\或()折行
```

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

```
print(is_leap)
```