

Python程序设计与案例教程

数据类型



数据类型

- 标识符、常量和变量
- python的基本数据类型
- 运算符与表达式
- math库及使用
- 数据类型转换



2.1 常量、变量与标识符

1. 标识符

- **标识符**在程序中用来标识各种程序成分，命名程序中的一**些实体**。包括**三方面**：
 - **1.字母； 2. 数字； 3. 下划线：“_”**
- **合法的标识符**：x, a1, wang, num_1, radius, **不合法的标识符**：first-name, 5sum

单独的下划线（_）用于表示上一次运算的结果

```
>>>2*3
```

```
6
```

```
>>>_*10
```

```
60
```



使用标识符应注意：

(1) 见名知义：通过变量名就知道变量值的含义

(2) 开头和结尾都使用下划线的情况应该避免：

sum

(3) 不能和关键字重名



Python中的33个关键字

```
>>>import keyword
```

```
>>>print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break',  
'class', 'continue', 'def', 'del', 'elif', 'else', 'except',  
'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is',  
'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',  
'return', 'try', 'while', 'with', 'yield']
```



keyword.iskeyword (word)

```
>>> keyword.iskeyword ('and')
```

True

```
>>> keyword.iskeyword ('int')
```

False



2. 常量

在程序运行过程中，其值不能改变的量称为常量。

(1) 整型常量： -123、20

(2) 实型常量： 3.14、0.15、-2.0

(3) 字符串常量： 'Python' 、 “Hello World! ”

(4) 布尔型常量： True、False

(5) 复数类型常量： 3+2.5j



2.1 常量、变量与标识符

3. 变量

在Python中，不需要事先声明变量名及其类型，类型是在运行过程中自动决定的，**直接赋值即可创建各种类型的变量。**

例如：

```
>>>x=5
```

```
>>>type(x)
```

```
<class 'int'>
```

采用内置函数`type()`返回变量`x`的类型`int`。



(1) type()函数

一般形式: **type(变量名)**

```
>>>x=5
```

```
>>>type(x)
```

```
<class 'int'>
```

```
>>>x="Hello World! "
```

```
>>>type(x)
```

```
<class 'str'>
```

```
>>>x=(1,2,3)
```

```
>>>type(x)
```

```
<class 'tuple'>
```

(2) id()函数

一般形式: **id(变量名)**

```
>>> str="Hello World!"
```

```
>>> id(str)
```

```
48768128
```



(3) isinstance ()函数

一般形式: **isinstance (对象, 类型名)**

```
>>> a=5
```

```
>>> isinstance(a,int)
```

```
True
```

```
>>> isinstance(a,str)
```

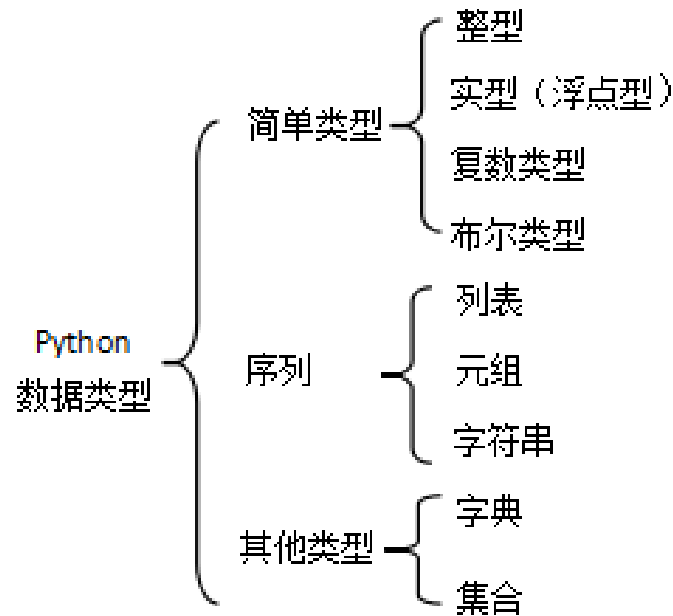
```
False
```

```
>>> isinstance(a,(float,int,str))
```

```
True
```

2.2 Python的基本数据类型

在程序设计语言中，都是采用数据类型来描述程序中的数据结构、数据的表示范围和数据在内存中的存储分配等。



整型数据

整型数据即整数，不带小数点，可以有正号或者负号。

在Python3.x中，整型数据在计算机内的表示没有长度限制，其值可以任意大。

```
>>> a=12345678900123456789
```

```
>>> b= a*a
```

```
>>> b
```

```
152415787504953525750053345778750190521
```



整型数据

- **十进制整数：**由0到9组成。
- **二进制整数：**以0b或0B为前缀，由0和1组成。
- **八进制整数：**以0o或0O为前缀，其后由0至7的数字组成。
- **十六进制整数：**以0x或0X开头，其后由0至9的数字和a至f字母或A至F字母组成。

根据前缀来区分各种进制数



实型数据

实数又称浮点数，一般有两种表示形式：

- **十进制小数形式：**由数字和小数点组成（必须有小数点）
- **指数形式：**用科学计数法表示的浮点数，用字母e（或E）表示以10为底的指数，e之前为数字部分，之后为指数部分。

注意：e（或E）前面必须有数字，后面必须是整数。

例如： e-5, 1.2E-3.5, 1e



字符型数据

字符型数据可以使用一对**单引号**、**双引号**或者**三引号**

```
>>> "Let's go"
```

```
"Let's go"
```

```
>>> s="'Python' Program"
```

```
>>> s
```

```
"'Python' Program"
```



字符型数据

转义字符

字符形式	含 义
<code>\n</code>	回车换行，将当前位置移到下一行开头
<code>\t</code>	横向跳到下一制表位置 (Tab)
<code>\b</code>	退格，将当前位置退回到前一列
<code>\r</code>	回车，将当前位置移到当前行开头
<code>\f</code>	走纸换页，将当前位置移到下页开头
<code>\\</code>	反斜线符“\”
<code>\'</code>	单引号符
<code>\"</code>	双引号符
<code>\ddd</code>	1~3位8进制数所代表的字符
<code>\xhh</code>	1~2位16进制数所代表的字符



转义字符

【例】 转义字符。

```
a=1
```

```
b=2
```

```
c='\101'
```

```
print("\t%d\n%d%s\n%d%d\t%s"%(a,b,c,a,b,c))
```

运行结果:

```
□□□□□□□1
```

```
2A
```

```
12□□□□□A
```



布尔型数据

Python的布尔类型有两个值：**True**和**False**，分别表示逻辑真和逻辑假。

```
>>> type(True)
```

```
<class 'bool'>
```

```
>>> True==1
```

```
True
```

```
>>> False==0
```

```
True
```

布尔型数据

布尔类型还可以与其他数据类型进行**逻辑**运算，Python规定：0、空字符串、None为False，其他数值和非空字符串为True。

```
>>> 0 and False
```

```
0
```

```
>>> None or True
```

```
True
```



复数类型数据

复数由两部分组成：实部和虚部

复数的形式为：实部+虚部j

```
>>> x=3+5j
```

```
>>> x.real
```

```
3.0
```

```
>>> x.imag
```

```
5.0
```

```
>>> y=6-10j
```

```
>>> x+y
```

```
(9-5j)
```

2.3 运算符与表达式

■ Python运算符

Python语言的运算符按照它们的功能可分为：

- (1) 算术运算符 (+、-、*、/、**、//、%) 。
- (2) 关系运算符 (>、<、>=、<=、==、!=) 。
- (3) 逻辑运算符 (and、or、not) 。
- (4) 位运算符 (<<、>>、~、|、^、&) 。
- (5) 赋值运算符 (=、复合赋值运算符) 。
- (6) 成员运算符 (in、not in) 。
- (7) 同一运算符 (is、is not) 。
- (8) 下标运算符 ([]) 。
- (9) 其他 (如函数调用运算符 ()) 。



2.3 运算符与表达式

几个相关概念

- ✓ 目（操作数个数）
- ✓ 优先级
- ✓ 结合性
- ✓ 表达式



2.3 运算符与表达式

1. 算术运算符和算术表达式

运算符	含义	优先级
+	加法	这些运算符的优先级相同， 但比下面的运算符优先级低
-	减法	
*	乘法	
/	除法	这些运算符的优先级相同， 但比上面的运算符优先级高
//	取整除	
**	幂运算	
%	取模	



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/388117053117006140>