

# 5.Python数字类型

## 学习目标

---

1. 了解编程语言中数据类型的本质
2. 熟悉基本的数字类型有哪些
3. 知道数据类型之间的转换
4. 知道如何将字符串转换为整数或浮点数
5. 知道如何查询数据类型相关的方法
6. 了解常用的内置数据类型函数和模块
7. 熟练使用常用的数学运算

## Python中所有类型的特点

---

如果大家详细回忆一下之前的课程就会发现了数据类型的定义

数据类型是一个值的集合以及定义在这个值集合上的一组操作的总称

发现了，就是说，所有的数据类型都是有对应的值表示的，那么这个值是什么，长什么样我们先不用关心，后面会慢慢学到。同时我们也知道了使用这个值，其实就是对这个值的操作。那么不同的数据类型可以进行的操作是不一样的，我们只需要知道这些数据类型都有哪些操作，以及如何去查看就行啦。不需要所有的操作都记住的，因为实在是太多了，而且很多我们平时写代码是根本用不上的。因此只需要记住那些常用的就行了。

## 数字类型分类

---

数字类型分为整型（int），浮点型（float）和复数类型（complex）。其实还有另外两个类型，一个是decimal（十进制），一个是fraction（分数）。

### 整数（整型）

所谓的整型数据其实就是我们数学所学的整数，整数包括了 **正数**、**负数** 和 **0**

比如：

```
10, 20, 30, -1, -100
```

以上这些都是整数，在编程里面叫整型或整数类型。

整数类型在Python里面用 `int` 来表示。`int` 在Python里面其实是一个类，我们可以通过在pycharm编辑器里面查看int的源码。

```
96
97 * class int(object):
98     """
99     int([x]) -> integer
100     int(x, base=10) -> integer
101
102     Convert a number or string to an integer, or return 0 if no arguments
103     are given. If x is a number, return x.__int__(). For floating point
104     numbers, this truncates towards zero.
105
106     If x is not a number or if base is given, then x must be a string,
107     bytes, or bytearray instance representing an integer literal in the
108     given base. The literal can be preceded by '+' or '-' and be surrounded
109     by whitespace. The base defaults to 10. Valid bases are 0 and 2-36.
110     Base 0 means to interpret the base from the string as an integer literal.
111     >>> int('0b100', base=0)
112     4
113     """
```

可以看到，这里有个 `class int(object)` 这个语法就是类的声明，后面我们学到面向对象的时候就会讲到。现在先大概了解一下，不需要记住。

一般情况下，对于非整数类型的数据，如果要转换为整数，就需要使用 `int` 类实例化，将对方的数据进行转换。

小技巧，如何查看一个数的数据类型？

使用type函数就可以查看啦，比如查看数字10是什么类型，那么就可以使用 `type(10)` 这样的方式进行查看。其它所有的类型也是如此，后面我们会有大量的练习来讲解，这里只需要记住就行了。

## 如何将其他的数据类型转换为 `int` 类型

使用int的实例将其它数据类型转换为整数。

这里有两种方式：

1. `int([x])`：这个使用方式的意思就是将数字或字符串转换为整数，如果什么都不填，那么就返回 `0`。
2. `int(x, base=10)`：这个就是将x转成10进制，x只能是数字或字符串、bytes类型或bytearray类型。字符串类型的允许前面加上正负号( `+-` )。

实例演示：

**将浮点数转换为整数**

```
x = int(10.3) # 这个会直接把后面的小数点给去掉，不会进行四舍五入，如果要四舍五入用round函数
y = round(10.6) # round函数会对里面的数字进行四舍五入
```

## 将字符串转换为整数

```
x = int('123') # 注意字符串里面只能是数字，不能包含字幕或标点符号等，否则会报错
y = int('-234') # 可以添加正负号在前
z = int(b'0123') # byte类型的字符串转换为整型
```

需要注意的是，如果字符串中的数值是其它进制的表示，则只能转换为对应进制的整数，否则会报错。对应的二进制的字符串表示是 `0b` 开头的，八进制的字符串表示是 `0o` 开头，十六进制的字符串表示是 `0x` 开头。

```
0b1010 # 这是一个二进制数，二进制数只能包含0和1两个数
0o1234 # 这是一个八进制数，八进制数是从0-7
0x123f # 这是一个十六进制数，十六进制的表示是0-9，a-f

# 其它进制转十进制
int(0b1010) # 二进制转十进制
int(0o1234) # 八进制转十进制
int(0x123f) # 十六进制转十进制

# 如果是字符串的数字，可以通过指定其进制数，然后转换成整数，
# 但是需要注意的是，字符串中的数字表示不能是非该进制中的数字或符号
int('10', base=2) # 指定为二进制转换
int('10', base=8) # 指定为八进制转换
int('10', base=16) # 指定为十六进制转换

# 字符串表示其它进制的，需要明确指定其进制数才能转换为整数
int('0b101', base=2) # 需制定是二进制
int('0o136', base=8) # 需指定是八进制
int('0xabc', base=16) # 需指定是十六进制
```

## 其它数据类型转换相关函数

```
hex(10) # 十进制转换为十六进制
oct(10) # 十进制转八进制
bin(10) # 十进制转二进制
```

## 浮点数（浮点型）

浮点数其实就是小数，凡是带有小数点的数都是小数。

浮点数在Python里面用float表示它的类型。但是实际上写的时候就是直接写字面的值就行。

比如：

```
1.23, 3.1415, 1.732, 100.00, -99.88
```

我们从源码也可以看到，float也是一个类型来的。

```
* class float(object):
    """ Convert a string or number to a floating point number, if possible. """
* def as_integer_ratio(self): # real signature unknown; restored from __doc__
    """
    Return integer ratio.

    Return a pair of integers, whose ratio is exactly equal to the original float
    and with a positive denominator.

    Raise OverflowError on infinities and a ValueError on NaNs.

    >>> (10.0).as_integer_ratio()
    (10, 1)
    >>> (0.0).as_integer_ratio()
    (0, 1)
    >>> (-.25).as_integer_ratio()
    (-1, 4)
```

## 数据类型转换

将其它数据类型转换为浮点数只有一种方法，就是通过float实例进行转换。

- float(x) : 这里的x可以是数字或字符串，字符串可以包含点，但是不能包含其他的非数字字符

实例演示：

```
float(0b101) # 二进制转浮点数
float(0o110) # 八进制转浮点数
float(10001) # 十进制转浮点数
float(0x110) # 十六进制转浮点数

float('100') # 将字符串转浮点数
float('3.1425') # 将字符串转浮点数，只能包含点
float('0x123') # 这种的是不行的，转换不了成浮点数
```

## 布尔类型

布尔类型 bool 在Python里面的表示就是 True 或 False 分别表示真和假。可以用数字1或0表示。只有数字0的条件判断是 False，其余非0的数字都表示真。

如何知道一个数字或其它类型值的布尔条件是真还是假呢？可以用 `bool` 实例进行验证。

```
bool(1)
bool(0)
bool(-100)
bool(3.14)
bool(0.0)
bool(-1.34)
```

至于其他的数据类型，等我们学到的时候会再次讲解。

## 运算符

---

运算符包括：

- 算术运算符
- 赋值运算符
- 比较运算符
- 逻辑运算符（不要求完全掌握）
- 位运算符（不需要掌握）
- 成员运算符（后面再学）
- 身份运算符（关系运算符）

## 算术运算符

单纯的一个数字在大多数情况下是没有什么特别大的使用价值的。我们运用数字往往需要是做某种运算。我们在小学就开始学习数学运算了。相信大家都还记得常用的运算法则。

**Python中常用的算术运算符及其表示：**

符号	说明
+	加号，表示两个数相加，比如： $x = 1 + 1$
-	减号，表示两个数相减，比如： $x = 2 - 1$
*	星号，表示两个数相乘，比如： $x = 3 * 2$
/	斜杠，表示两个数相除，比如： $x = 30 / 3$
%	百分号，表示求余数，比如： $x = 10 \% 3$
**	两个星号，表示幂运算，比如： $x = 2 ** 3$ ，这里就表示x等于2的3次方
//	两个斜杠，表示向下取整（地板除），比如： $x = 10 // 3$

## 赋值运算符

我们前面都只是对数字进行了运算，但是并没有保存运算的结果，那么如果要保存结果怎么办？或者是对于一些比较长一点的运算我们能不能简化？这就需要用到赋值运算符啦。

符号	说明
=	普通赋值等于号，让左值等于右值
-=	减号等于号，读作减等于，相当于 $x = x - n$
+=	加号等于号，读作加等于，相当于 $x = x + n$
*=	星号等于号，读作乘等于，相当于 $x = x * n$
/=	斜杠等于号，读作除等于，相当于 $x = x / n$
%=	百分号等于号，读作取余等于，相当于 $x = x \% n$
**=	两个星号等于号，读作幂等于，相当于 $x = x ** n$
//=	两个斜杠等于号，读作地板除等于，相当于 $x = x // n$
:=	冒号等于号，这个是海象运算符，是3.8版本才有的，可在表达式内部为变量赋值

## 比较运算符

比较运算符就是比较两个数的大小关系，常用的比较运算符有

符号	说明
>	大于号，判断左边的数是否大于右边的数
<	小于号，判断左边的数是否小于右边的数
!=	不等于,判断左边的数是否不等于右边的数
==	两个等于号，判断左边的数是否等于右边的数，求值得相等
>=	大于等于，判断左边的数是否大于或等于右边的数
<=	小于等于，判断左边的数是否小于或等于右边的数

## 逻辑运算符

符号	说明
and	表示逻辑与，只有两个条件都为真的时候才成立，在计算的时候，是False那么返回False所在边的值，如果是True返回右边的值
or	表示逻辑或，只要有一个为真条件就成立，在计算的时候，如果左边是True返回左边的值，如果左边是False则返回右边的值。
not	表示逻辑非，如果是False则返回True，如果是True则返回False

## 位运算符

位运算符主要就是对二进制的位进行逻辑运算的，这个只是做了解就行，不需要记住

符号	说明
&	按位与运算符：参与运算的两个值,如果两个相应位都为1,则该位的结果为1,否则为0
	按位或运算符：只要对应的二个二进位有一个为1时，结果位就为1。
^	按位异或运算符：当两对应的二进位相异时，结果为1
~	按位取反运算符：对数据的每个二进制位取反,即把1变为0,把0变为1。~x 类似于 -x-1
<<	左移动运算符：运算数的各二进位全部左移若干位，由"<<"右边的数指定移动的位数，高位丢弃，低位补0。
>>	右移动运算符：把">>"左边的运算数的各二进位全部右移若干位，">>"右边的数指定移动的位数

## 成员运算符

成员运算符主要是用于在序列中判断某个元素是否在序列里面。这个在后续的字符串、列表、元组、集合以及字典的课程中会使用到。在数字类型中是用不到的。

符号	说明
in	如果在指定的序列中找到值返回 True，否则返回 False。
not in	如果在指定的序列中没有找到值返回 True，否则返回 False。

## 身份运算符

身份运算符用于比较两个对象的存储单元，也叫关系运算符

运算符	描述
is	is 是判断两个标识符是不是引用自一个对象
is not	is not 是判断两个标识符是不是引用自不同对象

# 数字计算相关的函数和模块

## 相关函数



1. round函数, 对数字进行四舍五入
2. max函数, 求最大值
3. min函数求最小值
4. reduce函数, 累加计算, 这个函数在functools模块下
5. abs函数, 求绝对值
6. math数学模块
  - math.pow() 函数是幂运算
  - math.sqrt() 函数进行开根运算
7. random模块
  - random.randint() 生成制定范围的随机整数
  - random.random() 生成0-1之间的随机小数
  - random.randrange() 生成指定范围内的随机整数, 可以指定步长

## 作业

---

1. 尝试将以下字符串转换为整数或浮点数

```
a = '0x13f'
b = '0b10101'
c = '0xfff'
d = '0o773'
e = '100.23'
f = '0100.998'
g = '-0x136'
```

2. 分别将以下的数字转换为十六进制、八进制和二进制数的表达

```
a = 1024
b = 2480
c = -100
d = -10086
```

3. 如果发布文章的时候, 需要随机生成100-999之间的数值作为文章的阅读数应该如何实现? 写出你知道的方法, 并用代码实现出来。

要求只写出随机数生成方式即可, 可以有多种实现

3. 判断下列表达式的返回值分别是什么

```
x = 1 or 100
y = 0 and 10
z = 0 or 9
j = 100 and 0
i = not 100
a = False or True
b = True and 100
```

5. 判断下列的表达式是否成立

```
a = 100
b = 20
c = 100.01
a > b and b < c
a < b or c > b
a > b is True
a < b is True
c != a is False
b *= 6
b > a is True
```