

Python3 数字(Number)

Python 数字数据类型用于存储数值。

数据类型是不允许改变的,这就意味着如果改变数字数据类型的值,将重新分配内存空间。

以下实例在变量赋值时 Number 对象将被创建:

```
var1 = 1
var2 = 10
```

您也可以使用del语句删除一些数字对象的引用。

del语句的语法是:

```
del var1[,var2[,var3[...varN]]]
```

您可以通过使用del语句删除单个或多个对象的引用,例如:

```
del var
del var_a, var_b
```

Python 支持三种不同的数值类型:

- **整型(int)** - 通常被称为是整型或整数,是正或负整数,不带小数点。Python3 整型是没有限制大小的,可以当作 Long 类型使用,所以 Python3 没有 Python2 的 Long 类型。
- **浮点型(float)** - 浮点型由整数部分与小数部分组成,浮点型也可以使用科学计数法表示 ($2.5e2 = 2.5 \times 10^2 = 250$)
- **复数((complex))** - 复数由实数部分和虚数部分构成,可以用a + bj,或者complex(a,b)表示,复数的实部a和虚部b都是浮点型。

我们可以使用十六进制和八进制来代表整数:

```
>>> number = 0xA0F # 十六进制
>>> number
2575

>>> number=0o37 # 八进制
>>> number
31
```

int	float	complex
10	0.0	3.14j

100	15.20	45.j
-786	-21.9	9.322e-36j
080	32.3+e18	.876j
-0490	-90.	-.6545+0J
-0x260	-32.54e100	3e+26J
0x69	70.2-E12	4.53e-7j

- Python支持复数，复数由实数部分和虚数部分构成，可以用`a + bj`,或者`complex(a,b)`表示，复数的实部`a`和虚部`b`都是浮点型。

Python 数字类型转换

有时候，我们需要对数据内置的类型进行转换，数据类型的转换，你只需要将数据类型作为函数名即可。

- **int(x)** 将x转换为一个整数。
- **float(x)** 将x转换到一个浮点数。
- **complex(x)** 将x转换到一个复数，实数部分为 x，虚数部分为 0。
- **complex(x, y)** 将 x 和 y 转换到一个复数，实数部分为 x，虚数部分为 y。x 和 y 是数字表达式。

以下实例将浮点数变量 a 转换为整数：

```
>>> a = 1.0
>>> int(a)
1
```

Python 数字运算

Python 解释器可以作为一个简单的计算器，您可以在解释器里输入一个表达式，它将输出表达式的值。

表达式的语法很直白：`+`，`-`，`*` 和 `/`，和其它语言（如Pascal或C）里一样。例如：

```
>>> 2 + 2
4
>>> 50 - 5*6
20
>>> (50 - 5*6) / 4
5.0
```

```
>>> 8 / 5 # 总是返回一个浮点数
1.6
```

注意：在不同的机器上浮点运算的结果可能会不一样。

在整数除法中，除法 `/` 总是返回一个浮点数，如果只想得到整数的结果，丢弃可能的分数部分，可以使用运算符 `//`：

```
>>> 17 / 3 # 整数除法返回浮点型
5.666666666666667
>>>
>>> 17 // 3 # 整数除法返回向下取整后的结果
5
>>> 17 % 3 # %操作符返回除法的余数
2
>>> 5 * 3 + 2
17
```

注意：`//` 得到的并不一定是整数类型的数，它与分母分子的数据类型有关系。

```
>>> 7//2
3
>>> 7.0//2
3.0
>>> 7//2.0
3.0
>>>
```

等号 `=` 用于给变量赋值。赋值之后，除了下一个提示符，解释器不会显示任何结果。

```
>>> width = 20
>>> height = 5*9
>>> width * height
900
```

Python 可以使用 `**` 操作来进行幂运算：

```
>>> 5 ** 2 # 5 的平方
25
>>> 2 ** 7 # 2的7次方
128
```

变量在使用前必须先"定义"（即赋予变量一个值），否则会出现错误：

```
>>> n # 尝试访问一个未定义的变量
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
NameError: name 'n' is not defined
```

不同类型的数混合运算时会将整数转换为浮点数：

```
>>> 3 * 3.75 / 1.5
7.5
>>> 7.0 / 2
3.5
```

在交互模式中，最后被输出的表达式结果被赋值给变量 `_`。例如：

```
>>> tax = 12.5 / 100
>>> price = 100.50
>>> price * tax
12.5625
>>> price + _
113.0625
>>> round(_, 2)
113.06
```

此处，`_` 变量应被用户视为只读变量。

数学函数

函数	返回值（描述）
abs(x)	返回数字的绝对值，如abs(-10) 返回 10
ceil(x)	返回数字的上入整数，如math.ceil(4.1) 返回 5
cmp(x, y)	如果 x < y 返回 -1, 如果 x == y 返回 0, 如果 x > y 返回 1。 Python 3 已废弃 。使用 使用 (x>y)-(x<y) 替换。
exp(x)	返回e的x次幂(e ^x),如math.exp(1) 返回2.718281828459045
fabs(x)	返回数字的绝对值，如math.fabs(-10) 返回10.0
floor(x)	返回数字的下舍整数，如math.floor(4.9)返回 4
log(x)	如math.log(math.e)返回1.0,math.log(100,10)返回2.0
log10(x)	返回以10为基数的x的对数，如math.log10(100)返回 2.0
max(x1,	返回给定参数的最大值，参数可以为序列。

<code>x2,...)</code>	
<code>min(x1, x2,...)</code>	返回给定参数的最小值，参数可以为序列。
<code>modf(x)</code>	返回x的整数部分与小数部分，两部分的数值符号与x相同，整数部分以浮点型表示。
<code>pow(x,y)</code>	x^y 运算后的值。
<code>round(x[,n])</code>	返回浮点数x的四舍五入值，如给出n值，则代表舍入到小数点后的位数。
<code>sqrt(x)</code>	返回数字x的平方根。

随机数函数

随机数可以用于数学，游戏，安全等领域中，还经常被嵌入到算法中，用以提高算法效率，并提高程序的安全性。

Python包含以下常用随机数函数：

函数	描述
<code>choice(seq)</code>	从序列的元素中随机挑选一个元素，比如 <code>random.choice(range(10))</code> ，从0到9中随机挑选一个整数。
<code>randrange([start,] stop [,step])</code>	从指定范围内，按指定基数递增的集合中获取一个随机数，基数缺省值为1
<code>random()</code>	随机生成下一个实数，它在[0,1)范围内。
<code>seed([x])</code>	改变随机数生成器的种子seed。如果你不了解其原理，你不必特别去设定seed，Python会帮你选择seed。
<code>shuffle(lst)</code>	将序列的所有元素随机排序
<code>uniform(x,y)</code>	随机生成下一个实数，它在[x,y)范围内。

三角函数

Python包括以下三角函数：

函数	描述
<code>acos(x)</code>	返回x的反余弦弧度值。
<code>asin(x)</code>	返回x的正弦弧度值。
<code>atan(x)</code>	返回x的反正切弧度值。

<code>atan2(y,x)</code>	返回给定的 X 及 Y 坐标值的反正切值。
<code>cos(x)</code>	返回x的弧度的余弦值。
<code>hypot(x,y)</code>	返回欧几里德范数 $\sqrt{x^2 + y^2}$ 。
<code>sin(x)</code>	返回的x弧度的正弦值。
<code>tan(x)</code>	返回x弧度的正切值。
<code>degrees(x)</code>	将弧度转换为角度,如degrees(math.pi/2) , 返回90.0
<code>radians(x)</code>	将角度转换为弧度

数学常量

常量	描述
pi	数学常量 pi (圆周率，一般以 π 来表示)
e	数学常量 e，e即自然常数 (自然常数)。

+

14 篇笔记

写笔记