

# Python3 数字(Number)

---

## Python3 数字(Number)

[Python 数字类型转换](#)

[Python 数字运算](#)

[Python 随机数函数](#)

[random.random](#)

[random.uniform](#)

[random.randint](#)

[random.randrange](#)

[random.choice](#)

[random.shuffle](#)

[random.sample](#)

[数学函数](#)

[三角函数](#)

[数学常量](#)

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 x 100= 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。

---

## Python 随机数函数

---

### random.random

`random.random()`用于生成一个0到1的随机浮点数:  $0 \leq n < 1.0$

描述

**`random()`** 方法返回随机生成的一个实数，它在[0,1)范围内。

语法

以下是 `random()` 方法的语法：

```
import random

random.random()
```

**注意：**`random()`是不能直接访问的，需要导入 `random` 模块，然后通过 `random` 静态对象调用该方法。

参数

- 无

返回值

返回随机生成的一个实数，它在[0,1)范围内。

## 实例

以下展示了使用 random() 方法的实例：

```
#!/usr/bin/python3
import random

# 生成第一个随机数
print("random() : ", random.random())

# 生成第二个随机数
print("random() : ", random.random())
```

以上实例运行后输出结果为：

```
random() :  0.281954791393
random() :  0.309090465205
```

## random.uniform

random.uniform的函数原型为：random.uniform(a, b)，用于生成一个指定范围内的随机浮点数，两个参数其中一个为上限，一个为下限。如果  $a > b$ ，则生成的随机数  $n$ ：  $b \leq n \leq a$ 。如果  $a < b$ ，则  $a \leq n \leq b$ 。

```
print(random.uniform(10, 20))
print(random.uniform(20, 10))
```

结果（不同机器上的结果不一样）

13.01992427499013 11.6937509526014

## random.randint

random.randint()的函数原型为：random.randint(a, b)，用于生成一个指定范围内的整数。其中参数a是下限，参数b是上限，生成的随机数n：  $a \leq n \leq b$

```
#!/usr/bin/python3
import random

print(random.randint(12, 20)) #生成的随机数n： 12 <= n <= 20
print(random.randint(20, 20)) #结果永远是20
#print(random.randint(20, 10)) #该语句是错误的。下限必须小于上限。
```

## random.randrange

random.randrange的函数原型为：random.randrange([start], stop[, step])，从指定范围内，按指定基数递增的集合中 获取一个随机数。

如: random.randrange(10, 100, 2), 结果相当于从[10, 12, 14, 16, ... 96, 98]序列中获取一个随机数。  
random.randrange(10, 100, 2)在结果上与 random.choice(range(10, 100, 2)) 等效。

```
#!/usr/bin/python3
import random

print(random.randrange(10, 100, 2))
print(random.randrange( 1, 10, ))
```

## random.choice

random.choice从序列中获取一个随机元素。其函数原型为: random.choice(sequence)。参数sequence表示一个有序类型。这里要**说明**一下: sequence在python不是一种特定的类型, 而是泛指一系列的类型。list, tuple, 字符串都属于sequence。下面是使用choice的一些例子:

```
#!/usr/bin/python3
import random

print(random.choice("在联航学习Python"))
print(random.choice(["Tom", "is", "a", "handsome", "boy"]))
print(random.choice(("Tuple", "List", "Dict")))
```

## random.shuffle

random.shuffle的函数原型为: random.shuffle(x[, random]), 用于将一个列表中的元素打乱。如:

```
#!/usr/bin/python3
import random

p = ["Python", "is", "powerful", "simple", "and so on..."]
random.shuffle(p)
print(p)
```

## random.sample

random.sample的函数原型为: random.sample(sequence, k), 从指定序列中随机获取指定长度的片断。sample函数不会修改原有序列。

```
#!/usr/bin/python3
import random

list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
slice = random.sample(list, 5)      #从list中随机获取5个元素, 作为一个片断返回
print(slice)
print(list)                        #原有序列并没有改变。
```

---

## 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
exp(x)	返回e的x次幂(ex),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, x2,...)	返回给定参数的最大值, 参数可以为序列。
min(x1, x2,...)	返回给定参数的最小值, 参数可以为序列。
modf(x)	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。
pow(x, y)	$x^{**}y$ 运算后的值。
round(x [,n])	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。
sqrt(x)	返回数字x的平方根

## 三角函数

Python包括以下三角函数:

函数	描述
acos(x)	返回x的反余弦弧度值。
asin(x)	返回x的反正弦弧度值。
atan(x)	返回x的反正切弧度值。
atan2(y, x)	返回给定的 X 及 Y 坐标值的反正切值。
cos(x)	返回x的弧度的余弦值。
hypot(x, y)	返回欧几里德范数 $\sqrt{xx + yy}$ 。
sin(x)	返回的x弧度的正弦值。
tan(x)	返回x弧度的正切值。
degrees(x)	将弧度转换为角度,如degrees(math.pi/2) , 返回90.0
radians(x)	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 $\pi$ 来表示)
e	数学常量 e, e即自然常数 (自然常数) 。