# 1 Python Number(数字)

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

Python 支持四种不同的数值类型：

* **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。
* **长整型(long integers)** - 无限大小的整数，整数最后是一个大写或小写的L。
* **浮点型(floating point real values)** - 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 102 = 250）
* **复数( (complex numbers))** - 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 Number 类型转换

int(x [,base ]) 将x转换为一个整数 long(x [,base ]) 将x转换为一个长整数 float(x ) 将x转换到一个浮点数

complex(real [,imag ]) 创建一个复数

str(x ) 将对象 x 转换为字符串

repr(x ) 将对象 x 转换为表达式字符串 eval(str ) 用来计算在字符串中的有效Python表达式,并返回一个对象

tuple(s ) 将序列 s 转换为一个元组

list(s ) 将序列 s 转换为一个列表

chr(x ) 将一个整数转换为一个字符

unichr(x ) 将一个整数转换为Unicode字符

ord(x ) 将一个字符转换为它的整数值

hex(x ) 将一个整数转换为一个十六进制字符串

oct(x ) 将一个整数转换为一个八进制字符串

## Python数学函数

|  |  |
| --- | --- |
| **函数** | **返回值 ( 描述 )** |
| [abs(x)](http://www.runoob.com/python/func-number-abs.html) | 返回数字的绝对值，如abs(-10) 返回 10 |
| [ceil(x)](http://www.runoob.com/python/func-number-ceil.html) | 返回数字的上入整数，如math.ceil(4.1) 返回 5 |
| [cmp(x, y)](http://www.runoob.com/python/func-number-cmp.html) | 如果 x < y 返回 -1, 如果 x == y 返回 0, 如果 x > y 返回 1 |
| [exp(x)](http://www.runoob.com/python/func-number-exp.html) | 返回e的x次幂(ex),如math.exp(1) 返回2.718281828459045 |
| [fabs(x)](http://www.runoob.com/python/func-number-fabs.html) | 返回数字的绝对值，如math.fabs(-10) 返回10.0 |
| [floor(x)](http://www.runoob.com/python/func-number-floor.html) | 返回数字的下舍整数，如math.floor(4.9)返回 4 |
| [log(x)](http://www.runoob.com/python/func-number-log.html) | 如math.log(math.e)返回1.0,math.log(100,10)返回2.0 |
| [log10(x)](http://www.runoob.com/python/func-number-log10.html) | 返回以10为基数的x的对数，如math.log10(100)返回 2.0 |
| [max(x1, x2,...)](http://www.runoob.com/python/func-number-max.html) | 返回给定参数的最大值，参数可以为序列。 |
| [min(x1, x2,...)](http://www.runoob.com/python/func-number-min.html) | 返回给定参数的最小值，参数可以为序列。 |
| [modf(x)](http://www.runoob.com/python/func-number-modf.html) | 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)](http://www.runoob.com/python/func-number-pow.html) | x\*\*y 运算后的值。 |
| [round(x [,n])](http://www.runoob.com/python/func-number-round.html) | 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)](http://www.runoob.com/python/func-number-sqrt.html) | 返回数字x的平方根，数字可以为负数，返回类型为实数，如math.sqrt(4)返回 2+0j |

## Python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。

**import** random

Python包含以下常用随机数函数：

|  |  |
| --- | --- |
| **函数** | **描述** |
| [choice(seq)](http://www.runoob.com/python/func-number-choice.html) | 从序列的元素中随机挑选一个元素，比如random.choice(range(10))，  从0到9中随机挑选一个整数。 |
| [randrange ([start,] stop [,step])](http://www.runoob.com/python/func-number-randrange.html) | 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 |
| [random()](http://www.runoob.com/python/func-number-random.html) | 随机生成下一个实数，它在[0,1)范围内。 |
| [seed([x])](http://www.runoob.com/python/func-number-seed.html) | 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。 |
| [shuffle(lst)](http://www.runoob.com/python/func-number-shuffle.html) | 将序列的所有元素随机排序 |
| [uniform(x, y)](http://www.runoob.com/python/func-number-uniform.html) | 随机生成下一个实数，它在[x,y]范围内。 |
| random.randint(3,10) | 随机生成一个int数 |

生成随机验证码：

**import** random  
code=[]  
**for** x **in** range(6):  
 **if** x==random.randint(1,5):  
 code.append(str(random.randint(0,9)))  
 **else**:  
 code.append(chr(random.randint(65,90)))  
**print ''**.join(code)

## Python三角函数

Python包括以下三角函数：

|  |  |
| --- | --- |
| **函数** | **描述** |
| [acos(x)](http://www.runoob.com/python/func-number-acos.html) | 返回x的反余弦弧度值。 |
| [asin(x)](http://www.runoob.com/python/func-number-asin.html) | 返回x的反正弦弧度值。 |  |
| [atan(x)](http://www.runoob.com/python/func-number-atan.html) | 返回x的反正切弧度值。 |  |
| [atan2(y, x)](http://www.runoob.com/python/func-number-atan2.html) | 返回给定的 X 及 Y 坐标值的反正切值。 |  |
| [cos(x)](http://www.runoob.com/python/func-number-cos.html) | 返回x的弧度的余弦值。 |  |
| [hypot(x, y)](http://www.runoob.com/python/func-number-hypot.html) | 返回欧几里德范数 sqrt(x\*x + y\*y)。 |  |
| [sin(x)](http://www.runoob.com/python/func-number-sin.html) | 返回的x弧度的正弦值。 |  |
| [tan(x)](http://www.runoob.com/python/func-number-tan.html) | 返回x弧度的正切值。 |  |
| [degrees(x)](http://www.runoob.com/python/func-number-degrees.html) | 将弧度转换为角度,如degrees(math.pi/2) ， 返回90.0 |  |
| [radians(x)](http://www.runoob.com/python/func-number-radians.html) | 将角度转换为弧度 |  |

## Python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）。 |

# 2 Python 字符串

字符串是 Python 中最常用的数据类型。我们可以使用引号('或")来创建字符串。

## Python访问字符串中的值

Python不支持单字符类型，单字符也在Python也是作为一个字符串使用。

Python访问子字符串，可以使用方括号来截取字符串，如下实例：

#!/usr/bin/python

var1 = 'Hello World!'

var2 = "Python Runoob"

print "var1[0]: ", var1[0]print "var2[1:5]: ", var2[1:5]

以上实例执行结果：

var1[0]: H

var2[1:5]: ytho

## Python转义字符

在需要在字符中使用特殊字符时，python用反斜杠(\)转义字符。如下表：

| **转义字符** | **描述** |
| --- | --- |
| \(在行尾时) | 续行符 |
| \\ | 反斜杠符号 |
| \' | 单引号 |
| \" | 双引号 |
| \a | 响铃 |
| \b | 退格(Backspace) |
| \e | 转义 |
| \000 | 空 |
| \n | 换行 |
| \v | 纵向制表符 |
| \t | 横向制表符 |
| \r | 回车 |
| \f | 换页 |
| \oyy | 八进制数，yy代表的字符，例如：\o12代表换行 |
| \xyy | 十六进制数，yy代表的字符，例如：\x0a代表换行 |
| \other | 其它的字符以普通格式输出 |

## Python字符串运算符

下表实例变量 a 值为字符串 "Hello"，b 变量值为 "Python"：

|  |  |  |
| --- | --- | --- |
| **操作符** | **描述** | **实例** |
| + | 字符串连接 | >>>a + b 'HelloPython' |
| \* | 重复输出字符串 | >>>a \* 2 'HelloHello' |
| [] | 通过索引获取字符串中字符 | >>>a[1] 'e' |
| [ : ] | 截取字符串中的一部分 | >>>a[1:4] 'ell' |
| in | 成员运算符 - 如果字符串中包含给定的字符返回 True | >>>"H" in a True |
| not in | 成员运算符 - 如果字符串中不包含给定的字符返回 True | >>>"M" not in a True |
| r/R | 原始字符串 - 原始字符串：所有的字符串都是直接按照字面的意思来使用，没有转义特殊或不能打印的字符。 原始字符串除在字符串的第一个引号前加上字母"r"（可以大小写）以外，与普通字符串有着几乎完全相同的语法。 | >>>print r'\n' \n >>> print R'\n' \n |
| % | 格式字符串 | 请看下一章节 |

## Python字符串格式化

Python 支持格式化字符串的输出 。尽管这样可能会用到非常复杂的表达式，但最基本的用法是将一个值插入到一个有字符串格式符 %s 的字符串中。

在 Python 中，字符串格式化使用与 C 中 sprintf 函数一样的语法。

如下实例：

#!/usr/bin/python

print "My name is %s and weight is %d kg!" % ('Zara', 21)

以上实例输出结果：

My name is Zara and weight is 21 kg!

python字符串格式化符号:

<tbody

|  |  |
| --- | --- |
| **符   号** | **描述** |
| %c | 格式化字符及其ASCII码 |
| %s | 格式化字符串 |
| %d | 格式化整数 |
| %u | 格式化无符号整型 |
| %o | 格式化无符号八进制数 |
| %x | 格式化无符号十六进制数 |
| %X | 格式化无符号十六进制数（大写） |
| %f | 格式化浮点数字，可指定小数点后的精度 |
| %e | 用科学计数法格式化浮点数 |
| %E | 作用同%e，用科学计数法格式化浮点数 |
| %g | %f和%e的简写 |
| %G | %f 和 %E 的简写 |
| %p | 用十六进制数格式化变量的地址 |

格式化操作符辅助指令:

|  |  |
| --- | --- |
| **符号** | **功能** |
| \* | 定义宽度或者小数点精度 |
| - | 用做左对齐 |
| + | 在正数前面显示加号( + ) |
| <sp> | 在正数前面显示空格 |
| # | 在八进制数前面显示零('0')，在十六进制前面显示'0x'或者'0X'(取决于用的是'x'还是'X') |
| 0 | 显示的数字前面填充'0'而不是默认的空格 |
| % | '%%'输出一个单一的'%' |
| (var) | 映射变量(字典参数) |
| m.n. | m 是显示的最小总宽度,n 是小数点后的位数(如果可用的话) |

## Python三引号（triple quotes）

python中三引号可以将复杂的字符串进行复制:

python三引号允许一个字符串跨多行，字符串中可以包含换行符、制表符以及其他特殊字符。

可以用来输出xml 或者html

三引号的语法是一对连续的单引号或者双引号（通常都是成对的用）。

>>> hi = '''hi

there'''>>> hi # repr()'hi\nthere'>>> print hi # str()

hi

there

## Unicode 字符串

# 将Unicode转化为普通Python字符串："encode"   
utf8string = unicodestring.encode("utf-8")   
asciistring = unicodestring.encode("ascii")   
isostring = unicodestring.encode("ISO-8859-1")   
utf16string = unicodestring.encode("[utf-16](https://www.baidu.com/s?wd=utf-16&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3n1n1P1nkm1mLPAc4mWRk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHT1nWndP1nzn1R3P1f4n1b3Ps" \t "https://zhidao.baidu.com/question/_blank)")   
# 将普通Python字符串转化为Unicode："decode"   
plainstring1 = unicode(utf8string, "utf-8")   
plainstring2 = unicode(asciistring, "ascii")   
plainstring3 = unicode(isostring, "ISO-8859-1")   
plainstring4 = unicode(utf16string, "[utf-16](https://www.baidu.com/s?wd=utf-16&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3n1n1P1nkm1mLPAc4mWRk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHT1nWndP1nzn1R3P1f4n1b3Ps" \t "https://zhidao.baidu.com/question/_blank)")

Python 中定义一个 Unicode 字符串和定义一个普通字符串一样简单：

>>> u'Hello World !'

u'Hello World !'

引号前小写的"u"表示这里创建的是一个 Unicode 字符串。如果你想加入一个特殊字符，可以使用 Python 的 Unicode-Escape 编码。如下例所示：

>>> u'Hello\u0020World !'

u'Hello World !'

被替换的 \u0020 标识表示在给定位置插入编码值为 0x0020 的 Unicode 字符（空格符）。

## python的字符串内建函数

字符串方法是从python1.6到2.0慢慢加进来的——它们也被加到了Jython中。

这些方法实现了string模块的大部分方法，如下表所示列出了目前字符串内建支持的方法，所有的方法都包含了对Unicode的支持，有一些甚至是专门用于Unicode的。

|  |  |
| --- | --- |
| **方法** | **描述** |
| [string.capitalize()](http://www.runoob.com/python/att-string-capitalize.html" \t "http://www.runoob.com/python/_blank) | 把字符串的第一个字符大写 |
| [string.center(width)](http://www.runoob.com/python/att-string-center.html" \t "http://www.runoob.com/python/_blank) | 返回一个原字符串居中,并使用空格填充至长度 width 的新字符串 |
| [string.count(str, beg=0, end=len(string))](http://www.runoob.com/python/att-string-count.html" \t "http://www.runoob.com/python/_blank) | 返回 str 在 string 里面出现的次数，如果 beg 或者 end 指定则返回指定范围内 str 出现的次数 |
| [string.decode(encoding='UTF-8', errors='strict')](http://www.runoob.com/python/att-string-decode.html" \t "http://www.runoob.com/python/_blank) | 以 encoding 指定的编码格式解码 string，如果出错默认报一个 ValueError 的 异 常 ， 除 非 errors 指 定 的 是 'ignore' 或 者'replace' |
| [string.encode(encoding='UTF-8', errors='strict')](http://www.runoob.com/python/att-string-encode.html" \t "http://www.runoob.com/python/_blank) | 以 encoding 指定的编码格式编码 string，如果出错默认报一个ValueError 的异常，除非 errors 指定的是'ignore'或者'replace' |
| [string.endswith(obj, beg=0, end=len(string))](http://www.runoob.com/python/att-string-endswith.html" \t "http://www.runoob.com/python/_blank) | 检查字符串是否以 obj 结束，如果beg 或者 end 指定则检查指定的范围内是否以 obj 结束，如果是，返回 True,否则返回 False. |
| [string.expandtabs(tabsize=8)](http://www.runoob.com/python/att-string-expandtabs.html" \t "http://www.runoob.com/python/_blank) | 把字符串 string 中的 tab 符号转为空格，tab 符号默认的空格数是 8。 |
| [string.find(str, beg=0, end=len(string))](http://www.runoob.com/python/att-string-find.html" \t "http://www.runoob.com/python/_blank) | 检测 str 是否包含在 string 中，如果 beg 和 end 指定范围，则检查是否包含在指定范围内，如果是返回开始的索引值，否则返回-1 |
| [string.index(str, beg=0, end=len(string))](http://www.runoob.com/python/att-string-index.html" \t "http://www.runoob.com/python/_blank) | 跟find()方法一样，只不过如果str不在 string中会报一个异常. |
| [string.isalnum()](http://www.runoob.com/python/att-string-isalnum.html" \t "http://www.runoob.com/python/_blank) | 如果 string 至少有一个字符并且所有字符都是字母或数字则返  回 True,否则返回 False |
| [string.isalpha()](http://www.runoob.com/python/att-string-isalpha.html" \t "http://www.runoob.com/python/_blank) | 如果 string 至少有一个字符并且所有字符都是字母则返回 True,  否则返回 False |
| [string.isdecimal()](http://www.runoob.com/python/att-string-isdecimal.html" \t "http://www.runoob.com/python/_blank) | 如果 string 只包含十进制数字则返回 True 否则返回 False. |
| [string.isdigit()](http://www.runoob.com/python/att-string-isdigit.html" \t "http://www.runoob.com/python/_blank) | 如果 string 只包含数字则返回 True 否则返回 False. |
| [string.islower()](http://www.runoob.com/python/att-string-islower.html" \t "http://www.runoob.com/python/_blank) | 如果 string 中包含至少一个区分大小写的字符，并且所有这些(区分大小写的)字符都是小写，则返回 True，否则返回 False |
| [string.isnumeric()](http://www.runoob.com/python/att-string-isnumeric.html" \t "http://www.runoob.com/python/_blank) | 如果 string 中只包含数字字符，则返回 True，否则返回 False |
| [string.isspace()](http://www.runoob.com/python/att-string-isspace.html" \t "http://www.runoob.com/python/_blank) | 如果 string 中只包含空格，则返回 True，否则返回 False. |
| [string.istitle()](http://www.runoob.com/python/att-string-istitle.html" \t "http://www.runoob.com/python/_blank) | 如果 string 是标题化的(见 title())则返回 True，否则返回 False |
| [string.isupper()](http://www.runoob.com/python/att-string-isupper.html" \t "http://www.runoob.com/python/_blank) | 如果 string 中包含至少一个区分大小写的字符，并且所有这些(区分大小写的)字符都是大写，则返回 True，否则返回 False |
| [string.join(seq)](http://www.runoob.com/python/att-string-join.html" \t "http://www.runoob.com/python/_blank) | 以 string 作为分隔符，将 seq 中所有的元素(的字符串表示)合并为一个新的字符串 |
| [string.ljust(width)](http://www.runoob.com/python/att-string-ljust.html" \t "http://www.runoob.com/python/_blank) | 返回一个原字符串左对齐,并使用空格填充至长度 width 的新字符串 |
| [string.lower()](http://www.runoob.com/python/att-string-lower.html" \t "http://www.runoob.com/python/_blank) | 转换 string 中所有大写字符为小写. |
| [string.lstrip()](http://www.runoob.com/python/att-string-lstrip.html" \t "http://www.runoob.com/python/_blank) | 截掉 string 左边的空格 |
| [string.maketrans(intab, outtab])](http://www.runoob.com/python/att-string-maketrans.html" \t "http://www.runoob.com/python/_blank) | maketrans() 方法用于创建字符映射的转换表，对于接受两个参数的最简单的调用方式，第一个参数是字符串，表示需要转换的字符，第二个参数也是字符串表示转换的目标。 |
| [max(str)](http://www.runoob.com/python/att-string-max.html" \t "http://www.runoob.com/python/_blank) | 返回字符串 str 中最大的字母。 |
| [min(str)](http://www.runoob.com/python/att-string-min.html" \t "http://www.runoob.com/python/_blank) | 返回字符串 str 中最小的字母。 |
| [string.partition(str)](http://www.runoob.com/python/att-string-partition.html" \t "http://www.runoob.com/python/_blank) | 有点像 find()和 split()的结合体,从 str 出现的第一个位置起,把 字 符 串 string 分 成 一 个 3 元 素 的 元 组 (string\_pre\_str,str,string\_post\_str),如果 string 中不包含str 则 string\_pre\_str == string. |
| [string.replace(str1, str2,  num=string.count(str1))](http://www.runoob.com/python/att-string-replace.html" \t "http://www.runoob.com/python/_blank) | 把 string 中的 str1 替换成 str2,如果 num 指定，则替换不超过 num 次. |
| [string.rfind(str, beg=0,end=len(string) )](http://www.runoob.com/python/att-string-rfind.html" \t "http://www.runoob.com/python/_blank) | 类似于 find()函数，不过是从右边开始查找. |
| [string.rindex( str, beg=0,end=len(string))](http://www.runoob.com/python/att-string-rindex.html" \t "http://www.runoob.com/python/_blank) | 类似于 index()，不过是从右边开始. |
| [string.rjust(width)](http://www.runoob.com/python/att-string-rjust.html" \t "http://www.runoob.com/python/_blank) | 返回一个原字符串右对齐,并使用空格填充至长度 width 的新字符串 |
| string.rpartition(str) | 类似于 partition()函数,不过是从右边开始查找. |
| [string.rstrip()](http://www.runoob.com/python/att-string-rstrip.html" \t "http://www.runoob.com/python/_blank) | 删除 string 字符串末尾的空格. |
| [string.split(str="", num=string.count(str))](http://www.runoob.com/python/att-string-split.html" \t "http://www.runoob.com/python/_blank) | 以 str 为分隔符切片 string，如果 num有指定值，则仅分隔 num 个子字符串 |
| [string.splitlines(num=string.count('\n'))](http://www.runoob.com/python/att-string-splitlines.html" \t "http://www.runoob.com/python/_blank) | 按照行分隔，返回一个包含各行作为元素的列表，如果 num 指定则仅切片 num 个行. |
| [string.startswith(obj, beg=0,end=len(string))](http://www.runoob.com/python/att-string-startswith.html" \t "http://www.runoob.com/python/_blank) | 检查字符串是否是以 obj 开头，是则返回 True，否则返回 False。如果beg 和 end 指定值，则在指定范围内检查. |
| [string.strip([obj])](http://www.runoob.com/python/att-string-strip.html" \t "http://www.runoob.com/python/_blank) | 在 string 上执行 lstrip()和 rstrip() |
| [string.swapcase()](http://www.runoob.com/python/att-string-swapcase.html" \t "http://www.runoob.com/python/_blank) | 翻转 string 中的大小写 |
| [string.title()](http://www.runoob.com/python/att-string-title.html" \t "http://www.runoob.com/python/_blank) | 返回"标题化"的 string,就是说所有单词都是以大写开始，其余字母均为小写(见 istitle()) |
| [string.translate(str, del="")](http://www.runoob.com/python/att-string-translate.html" \t "http://www.runoob.com/python/_blank) | 根据 str 给出的表(包含 256 个字符)转换 string 的字符,  要过滤掉的字符放到 del 参数中 |
| [string.upper()](http://www.runoob.com/python/att-string-upper.html" \t "http://www.runoob.com/python/_blank) | 转换 string 中的小写字母为大写 |
| [string.zfill(width)](http://www.runoob.com/python/att-string-zfill.html" \t "http://www.runoob.com/python/_blank) | 返回长度为 width 的字符串，原字符串 string 右对齐，前面填充0 |
| [string.isdecimal()](http://www.runoob.com/python/att-string-isdecimal.html" \t "http://www.runoob.com/python/_blank) | isdecimal()方法检查字符串是否只包含十进制字符。这种方法只存在于unicode对象。 |

# 3 Python 列表(List)

序列是Python中最基本的数据结构。序列中的每个元素都分配一个数字 - 它的位置，或索引，第一个索引是0，第二个索引是1，依此类推。

序列都可以进行的操作包括索引，切片，加，乘，检查成员。

此外，Python已经内置确定序列的长度以及确定最大和最小的元素的方法。

列表的数据项不需要具有相同的类型

## 访问列表中的值

使用下标索引来访问列表中的值，同样你也可以使用方括号的形式截取字符，如下所示：

#!/usr/bin/python

list1 = ['physics', 'chemistry', 1997, 2000];

list2 = [1, 2, 3, 4, 5, 6, 7 ];

print "list1[0]: ", list1[0]

print "list2[1:5]: ", list2[1:5]

以上实例输出结果：

list1[0]: physics

list2[1:5]: [2, 3, 4, 5]

## 更新列表

你可以对列表的数据项进行修改或更新，你也可以使用append()方法来添加列表项，如下所示：

#!/usr/bin/python

list = ['physics', 'chemistry', 1997, 2000];

print "Value available at index 2 : "

print list[2];

list[2] = 2001;

print "New value available at index 2 : "

print list[2];

## 删除列表元素

可以使用 del 语句来删除列表的的元素：del list1[2];

## Python列表脚本操作符

列表对 + 和 \* 的操作符与字符串相似。+ 号用于组合列表，\* 号用于重复列表。

如下所示：

|  |  |  |
| --- | --- | --- |
| **Python 表达式** | **结果** | **描述** |
| len([1, 2, 3]) | 3 | 长度 |
| [1, 2, 3] + [4, 5, 6] | [1, 2, 3, 4, 5, 6] | 组合 |
| ['Hi!'] \* 4 | ['Hi!', 'Hi!', 'Hi!', 'Hi!'] | 重复 |
| 3 in [1, 2, 3] | True | 元素是否存在于列表中 |
| for x in [1, 2, 3]: print x, | 1 2 3 | 迭代 |

## Python列表截取

Python的列表截取与字符串操作类型，如下所示：

L = ['spam', 'Spam', 'SPAM!']

操作：

|  |  |  |
| --- | --- | --- |
| **Python 表达式** | **结果** | **描述** |
| L[2] | 'SPAM!' | 读取列表中第三个元素 |
| L[-2] | 'Spam' | 读取列表中倒数第二个元素 |
| L[1:] | ['Spam', 'SPAM!'] | 从第二个元素开始截取列表 |

## Python列表函数&方法

Python包含以下函数:

|  |  |
| --- | --- |
| **序号** | **函数** |
| 1 | [cmp(list1, list2)](http://www.runoob.com/python/att-list-cmp.html) 比较两个列表的元素 |
| 2 | [len(list)](http://www.runoob.com/python/att-list-len.html) 列表元素个数 |
| 3 | [max(list)](http://www.runoob.com/python/att-list-max.html) 返回列表元素最大值 |
| 4 | [min(list)](http://www.runoob.com/python/att-list-min.html) 返回列表元素最小值 |
| 5 | [list(seq)](http://www.runoob.com/python/att-list-list.html) 将元组转换为列表 |

Python包含以下方法:

|  |  |
| --- | --- |
| **序号** | **方法** |
| 1 | [list.append(obj)](http://www.runoob.com/python/att-list-append.html) 在列表末尾添加新的对象 |
| 2 | [list.count(obj)](http://www.runoob.com/python/att-list-count.html) 统计某个元素在列表中出现的次数 |
| 3 | [list.extend(seq)](http://www.runoob.com/python/att-list-extend.html) 在列表末尾一次性追加另一个序列中的多个值（用新列表扩展原来的列表） |
| 4 | [list.index(obj)](http://www.runoob.com/python/att-list-index.html) 从列表中找出某个值第一个匹配项的索引位置 |
| 5 | [list.insert(index, obj)](http://www.runoob.com/python/att-list-insert.html) 将对象插入列表 |
| 6 | [list.pop(obj=list[-1])](http://www.runoob.com/python/att-list-pop.html) 移除列表中的一个元素（默认最后一个元素），并且返回该元素的值 |
| 7 | [list.remove(obj)](http://www.runoob.com/python/att-list-remove.html) 移除列表中某个值的第一个匹配项 |
| 8 | [list.reverse()](http://www.runoob.com/python/att-list-reverse.html) 反向列表中元素 |
| 9 | [list.sort([func])](http://www.runoob.com/python/att-list-sort.html) 对原列表进行排序 |

# 4 Python 元组

Python的元组与列表类似，不同之处在于元组的元素不能修改。

元组使用小括号，列表使用方括号。

用法与列表差不多：删除后再调用会报错，不能修改，元祖相加会产生新的元祖

## 无关闭分隔符

任意无符号的对象，以逗号隔开，默认为元组，如下实例：

#!/usr/bin/python

print 'abc', -4.24e93, 18+6.6j, 'xyz';

x, y = 1, 2;

print "Value of x , y : ", x,y;

以上实例运行结果：

abc -4.24e+93 (18+6.6j) xyz Value of x , y : 1 2

# 5 Python 字典(Dictionary)

字典是另一种可变容器模型，且可存储任意类型对象。

## 访问字典里的值

如果用字典里没有的键访问数据，会输出错误如下：

把相应的键放入熟悉的方括弧，如下实例:

#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'};

print "dict['Name']: ", dict['Name'];print "dict['Age']: ", dict['Age'];

## 修改字典

向字典添加新内容的方法是增加新的键/值对，修改或删除已有键/值对如下实例:

#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'};

dict['Age'] = 8; # update existing entry

dict['School'] = "DPS School"; # Add new entry

print "dict['Age']: ", dict['Age'];print "dict['School']: ", dict['School'];

以上实例输出结果：

dict['Age']: 8

dict['School']: DPS School

## 删除字典元素

能删单一的元素也能清空字典，清空只需一项操作。

显示删除一个字典用del命令，如下实例：

#!/usr/bin/python# -\*- coding: UTF-8 -\*-

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'};

del dict['Name']; # 删除键是'Name'的条目

dict.clear(); # 清空词典所有条目

del dict ; # 删除词典

### **字典键的特性**

字典值可以没有限制地取任何python对象，既可以是标准的对象，也可以是用户定义的，但键不行。

两个重要的点需要记住：

1）不允许同一个键出现两次。创建时如果同一个键被赋值两次，后一个值会被记住，如下实例：

#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Name': 'Manni'};

print "dict['Name']: ", dict['Name'];

以上实例输出结果：

dict['Name']: Manni

2）键必须不可变，所以可以用数字，字符串或元组充当，所以用列表就不行，

## 字典内置函数&方法

Python字典包含了以下内置函数：

|  |  |
| --- | --- |
| **序号** | **函数及描述** |
| 1 | [cmp(dict1, dict2)](http://www.runoob.com/python/att-dictionary-cmp.html" \t "http://www.runoob.com/python/_blank) 比较两个字典元素。 |
| 2 | [len(dict)](http://www.runoob.com/python/att-dictionary-len.html" \t "http://www.runoob.com/python/_blank) 计算字典元素个数，即键的总数。 |
| 3 | [str(dict)](http://www.runoob.com/python/att-dictionary-str.html" \t "http://www.runoob.com/python/_blank) 输出字典可打印的字符串表示。 |
| 4 | [type(variable)](http://www.runoob.com/python/att-dictionary-type.html" \t "http://www.runoob.com/python/_blank) 返回输入的变量类型，如果变量是字典就返回字典类型。 |

Python字典包含了以下内置方法：

|  |  |
| --- | --- |
| **序号** | **函数及描述** |
| 1 | [radiansdict.clear()](http://www.runoob.com/python/att-dictionary-clear.html" \t "http://www.runoob.com/python/_blank) 删除字典内所有元素 |
| 2 | [radiansdict.copy()](http://www.runoob.com/python/att-dictionary-copy.html" \t "http://www.runoob.com/python/_blank) 返回一个字典的浅复制 |
| 3 | [radiansdict.fromkeys()](http://www.runoob.com/python/att-dictionary-fromkeys.html" \t "http://www.runoob.com/python/_blank) 创建一个新字典，以序列seq中元素做字典的键，val为字典所有键对应的初始值 |
| 4 | [radiansdict.get(key, default=None)](http://www.runoob.com/python/att-dictionary-get.html" \t "http://www.runoob.com/python/_blank) 返回指定键的值，如果值不在字典中返回default值 |
| 5 | [radiansdict.has\_key(key)](http://www.runoob.com/python/att-dictionary-has_key.html" \t "http://www.runoob.com/python/_blank) 如果键在字典dict里返回true，否则返回false |
| 6 | [radiansdict.items()](http://www.runoob.com/python/att-dictionary-items.html" \t "http://www.runoob.com/python/_blank) 以列表返回可遍历的(键, 值) 元组数组 |
| 7 | [radiansdict.keys()](http://www.runoob.com/python/att-dictionary-keys.html" \t "http://www.runoob.com/python/_blank) 以列表返回一个字典所有的键 |
| 8 | [radiansdict.setdefault(key, default=None)](http://www.runoob.com/python/att-dictionary-setdefault.html" \t "http://www.runoob.com/python/_blank) 和get()类似, 但如果键不存在于字典中，将会添加键并将值设为default |
| 9 | [radiansdict.update(dict2)](http://www.runoob.com/python/att-dictionary-update.html" \t "http://www.runoob.com/python/_blank) 把字典dict2的键/值对更新到dict里 |
| 10 | [radiansdict.values()](http://www.runoob.com/python/att-dictionary-values.html" \t "http://www.runoob.com/python/_blank) 以列表返回字典中的所有值 |

# 6 Python 日期和时间

Python 程序能用很多方式处理日期和时间，转换日期格式是一个常见的功能。

Python 提供了一个 time 和 calendar 模块可以用于格式化日期和时间。

时间间隔是以秒为单位的浮点小数。

每个时间戳都以自从1970年1月1日午夜（历元）经过了多长时间来表示。

Python 的 time 模块下有很多函数可以转换常见日期格式。如函数time.time()用于获取当前时间戳, 如下实例:

#!/usr/bin/python# -\*- coding: UTF-8 -\*-

import time; # 引入time模块

ticks = time.time()print "当前时间戳为:", ticks

以上实例输出结果：

当前时间戳为: 1459994552.51

## 什么是时间元组？

上述也就是struct\_time元组。这种结构具有如下属性：

|  |  |  |
| --- | --- | --- |
| **序号** | **属性** | **值** |
| 0 | tm\_year | 2008 |
| 1 | tm\_mon | 1 到 12 |
| 2 | tm\_mday | 1 到 31 |
| 3 | tm\_hour | 0 到 23 |
| 4 | tm\_min | 0 到 59 |
| 5 | tm\_sec | 0 到 61 (60或61 是闰秒) |
| 6 | tm\_wday | 0到6 (0是周一) |
| 7 | tm\_yday | 1 到 366(儒略历) |
| 8 | tm\_isdst | -1, 0, 1, -1是决定是否为夏令时的旗帜 |

## 获取当前时间

从返回浮点数的时间辍方式向时间元组转换，只要将浮点数传递给如localtime之类的函数。

#!/usr/bin/python# -\*- coding: UTF-8 -\*-

import time

localtime = time.localtime(time.time())print "本地时间为 :", localtime

**print** time.gmtime() #输出结果同上

以上实例输出结果：

本地时间为 : time.struct\_time(tm\_year=2016, tm\_mon=4, tm\_mday=7, tm\_hour=10, tm\_min=3, tm\_sec=27, tm\_wday=3, tm\_yday=98, tm\_isdst=0)

## 获取格式化的时间

你可以根据需求选取各种格式，但是最简单的获取可读的时间模式的函数是asctime():

#!/usr/bin/python# -\*- coding: UTF-8 -\*-

import time

localtime = time.asctime( time.localtime(time.time()) )print "本地时间为 :", localtime

以上实例输出结果：

本地时间为 : Thu Apr 7 10:05:21 2016

## 格式化日期

我们可以使用 time 模块的 strftime 方法来格式化日期，：

time.strftime(format[, t])

#!/usr/bin/python# -\*- coding: UTF-8 -\*-

import time

# 格式化成2016-03-20 11:45:39形式print time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())

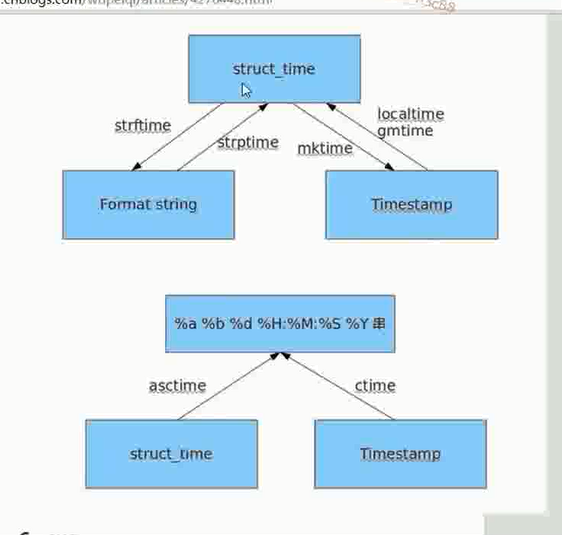
# 格式化成Sat Mar 28 22:24:24 2016形式print time.strftime("%a %b %d %H:%M:%S %Y", time.localtime())

# 将格式字符串转换为时间戳

a = "Sat Mar 28 22:24:24 2016"print time.mktime(time.strptime(a,"%a %b %d %H:%M:%S %Y"))

以上实例输出结果：

2016-04-07 10:25:09Thu Apr 07 10:25:09 20161459175064.0



python中时间日期格式化符号：

* %y 两位数的年份表示（00-99）
* %Y 四位数的年份表示（000-9999）
* %m 月份（01-12）
* %d 月内中的一天（0-31）
* %H 24小时制小时数（0-23）
* %I 12小时制小时数（01-12）
* %M 分钟数（00=59）
* %S 秒（00-59）
* %a 本地简化星期名称
* %A 本地完整星期名称
* %b 本地简化的月份名称
* %B 本地完整的月份名称
* %c 本地相应的日期表示和时间表示
* %j 年内的一天（001-366）
* %p 本地A.M.或P.M.的等价符
* %U 一年中的星期数（00-53）星期天为星期的开始
* %w 星期（0-6），星期天为星期的开始
* %W 一年中的星期数（00-53）星期一为星期的开始
* %x 本地相应的日期表示
* %X 本地相应的时间表示
* %Z 当前时区的名称
* %% %号本身

## 获取某月日历

Calendar模块有很广泛的方法用来处理年历和月历，例如打印某月的月历：

#!/usr/bin/python# -\*- coding: UTF-8 -\*-

import calendar

cal = calendar.month(2016, 1)print "以下输出2016年1月份的日历:"print cal;

以上实例输出结果：

以下输出2016年1月份的日历:

January 2016Mo Tu We Th Fr Sa Su

1 2 3

4 5 6 7 8 9 1011 12 13 14 15 16 1718 19 20 21 22 23 2425 26 27 28 29 30 31

## Time 模块

Time 模块包含了以下内置函数，既有时间处理相的，也有转换时间格式的：

|  |  |
| --- | --- |
| **序号** | **函数及描述** |
| 1 | [time.altzone](http://www.runoob.com/python/att-time-altzone.html" \t "http://www.runoob.com/python/_blank) 返回格林威治西部的夏令时地区的偏移秒数。如果该地区在格林威治东部会返回负值（如西欧，包括英国）。对夏令时启用地区才能使用。 |
| 2 | [time.asctime([tupletime])](http://www.runoob.com/python/att-time-asctime.html" \t "http://www.runoob.com/python/_blank) 接受时间元组并返回一个可读的形式为"Tue Dec 11 18:07:14 2008"（2008年12月11日 周二18时07分14秒）的24个字符的字符串。 |
| 3 | [time.clock( )](http://www.runoob.com/python/att-time-clock.html" \t "http://www.runoob.com/python/_blank) 用以浮点数计算的秒数返回当前的CPU时间。用来衡量不同程序的耗时，比time.time()更有用。 |
| 4 | [time.ctime([secs])](http://www.runoob.com/python/att-time-ctime.html" \t "http://www.runoob.com/python/_blank) 作用相当于asctime(localtime(secs))，未给参数相当于asctime() |
| 5 | [time.gmtime([secs])](http://www.runoob.com/python/att-time-gmtime.html" \t "http://www.runoob.com/python/_blank) 接收时间辍（1970纪元后经过的浮点秒数）并返回格林威治天文时间下的时间元组t。注：t.tm\_isdst始终为0 |
| 6 | [time.localtime([secs])](http://www.runoob.com/python/att-time-localtime.html" \t "http://www.runoob.com/python/_blank) 接收时间辍（1970纪元后经过的浮点秒数）并返回当地时间下的时间元组t（t.tm\_isdst可取0或1，取决于当地当时是不是夏令时）。 |
| 7 | [time.mktime(tupletime)](http://www.runoob.com/python/att-time-mktime.html" \t "http://www.runoob.com/python/_blank) 接受时间元组并返回时间辍（1970纪元后经过的浮点秒数）。 |
| 8 | [time.sleep(secs)](http://www.runoob.com/python/att-time-sleep.html" \t "http://www.runoob.com/python/_blank) 推迟调用线程的运行，secs指秒数。 |
| 9 | [time.strftime(fmt[,tupletime])](http://www.runoob.com/python/att-time-strftime.html" \t "http://www.runoob.com/python/_blank) 接收以时间元组，并返回以可读字符串表示的当地时间，格式由fmt决定。 |
| 10 | [time.strptime(str,fmt='%a %b %d %H:%M:%S %Y')](http://www.runoob.com/python/att-time-strptime.html" \t "http://www.runoob.com/python/_blank) 根据fmt的格式把一个时间字符串解析为时间元组。 |
| 11 | [time.time( )](http://www.runoob.com/python/att-time-time.html" \t "http://www.runoob.com/python/_blank) 返回当前时间的时间戳（1970纪元后经过的浮点秒数）。 |
| 12 | [time.tzset()](http://www.runoob.com/python/att-time-tzset.html" \t "http://www.runoob.com/python/_blank) 根据环境变量TZ重新初始化时间相关设置。 |

Time模块包含了以下2个非常重要的属性：

|  |  |
| --- | --- |
| **序号** | **属性及描述** |
| 1 | **time.timezone** 属性time.timezone是当地时区（未启动夏令时）距离格林威治的偏移秒数（>0，美洲;<=0大部分欧洲，亚洲，非洲）。 |
| 2 | **time.tzname** 属性time.tzname包含一对根据情况的不同而不同的字符串，分别是带夏令时的本地时区名称，和不带的。 |

## 日历（Calendar）模块

此模块的函数都是日历相关的，例如打印某月的字符月历。

星期一是默认的每周第一天，星期天是默认的最后一天。更改设置需调用calendar.setfirstweekday()函数。模块包含了以下内置函数：

|  |  |
| --- | --- |
| **序号** | **函数及描述** |
| 1 | **calendar.calendar(year,w=2,l=1,c=6)** 返回一个多行字符串格式的year年年历，3个月一行，间隔距离为c。 每日宽度间隔为w字符。每行长度为21\* W+18+2\* C。l是每星期行数。 |
| 2 | **calendar.firstweekday( )** 返回当前每周起始日期的设置。默认情况下，首次载入caendar模块时返回0，即星期一。 |
| 3 | **calendar.isleap(year)** 是闰年返回True，否则为false。 |
| 4 | **calendar.leapdays(y1,y2)** 返回在Y1，Y2两年之间的闰年总数。 |
| 5 | **calendar.month(year,month,w=2,l=1)** 返回一个多行字符串格式的year年month月日历，两行标题，一周一行。每日宽度间隔为w字符。每行的长度为7\* w+6。l是每星期的行数。 |
| 6 | **calendar.monthcalendar(year,month)** 返回一个整数的单层嵌套列表。每个子列表装载代表一个星期的整数。Year年month月外的日期都设为0;范围内的日子都由该月第几日表示，从1开始。 |
| 7 | **calendar.monthrange(year,month)** 返回两个整数。第一个是该月的星期几的日期码，第二个是该月的日期码。日从0（星期一）到6（星期日）;月从1到12。 |
| 8 | **calendar.prcal(year,w=2,l=1,c=6)** 相当于 print calendar.calendar(year,w,l,c). |
| 9 | **calendar.prmonth(year,month,w=2,l=1)** 相当于 print calendar.calendar（year，w，l，c）。 |
| 10 | **calendar.setfirstweekday(weekday)** 设置每周的起始日期码。0（星期一）到6（星期日）。 |
| 11 | **calendar.timegm(tupletime)** 和time.gmtime相反：接受一个时间元组形式，返回该时刻的时间辍（1970纪元后经过的浮点秒数）。 |
| 12 | **calendar.weekday(year,month,day)** 返回给定日期的日期码。0（星期一）到6（星期日）。月份为 1（一月） 到 12（12月）。 |

## 其他相关模块和函数

在Python中，其他处理日期和时间的模块还有：

* [datetime模块](http://docs.python.org/library/datetime.html" \l "module-datetime" \t "http://www.runoob.com/python/_blank)
* [pytz模块](http://www.twinsun.com/tz/tz-link.htm" \t "http://www.runoob.com/python/_blank)
* [dateutil模块](http://labix.org/python-dateutil" \t "http://www.runoob.com/python/_blank)