**数学运算函数：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [abs()](http://www.cnblogs.com/mywood/p/7429135.html#abs) | 返回数字的绝对值 | [divmod()](http://www.cnblogs.com/mywood/p/7429135.html" \l "divmod" \t "_blank) | 计算两个数的商和余数，用元组返回 | [pow()](http://www.cnblogs.com/mywood/p/7429135.html" \l "pow" \t "_blank) | 返回 xy（x的y次方） 的值 |
| [sum()](http://www.cnblogs.com/mywood/p/7429135.html#sum) | 求和计算。 | [bin()](http://www.cnblogs.com/mywood/p/7429135.html#bin) | 返回一个整数 int 或者长整数 long int 的二进制表示 | [oct()](http://www.cnblogs.com/mywood/p/7429135.html" \l "oct" \t "_blank) | 将一个整数转换成8进制字符串。 |
| [round()](http://www.cnblogs.com/mywood/p/7429135.html#round) | 返回浮点数x的四舍五入值。 | [long()](http://www.cnblogs.com/mywood/p/7429135.html#long) | 将数字或字符串转换为一个长整型 | [hex()](http://www.cnblogs.com/mywood/p/7429135.html#hex) | 将10进制整数转换成16进制整数。 |

**类型转换函数：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [str()](http://www.cnblogs.com/mywood/p/7429135.html#str) | 函数将对象转化为String的形式。 | [int()](http://www.cnblogs.com/mywood/p/7429135.html" \l "int" \t "_blank) | 函数用于将一个字符串会数字转换为整型。 | [tuple()](http://www.cnblogs.com/mywood/p/7429135.html#tuple) | 将列表转换为元组。 |
| [bool()](http://www.cnblogs.com/mywood/p/7429135.html#bool) | 参数转换为布尔类型，如果没有参数，返回 False | [float()](http://www.cnblogs.com/mywood/p/7429135.html#float) | 将整数和字符串转换成浮点数。 | [list()](http://www.cnblogs.com/mywood/p/7429135.html#list) | 将元组转换为列表 |

**序列操作函数：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [min()](http://www.cnblogs.com/mywood/p/7429135.html#min) | 返回给定参数的最小值，参数可以为序列。 | [sorte()](http://www.cnblogs.com/mywood/p/7429135.html" \l "sorted" \t "_blank) | 对所有可迭代的对象进行排序操作。 | [len()](http://www.cnblogs.com/mywood/p/7429135.html" \l "len" \t "_blank) | 返回对象（字符、列表、元组等）长度或项目个数 |
| [filter()](http://www.cnblogs.com/mywood/p/7429135.html#filter) | 用于过滤序列，过滤掉不符合条件的元素，返回由符合条件元素组成的新列表 | | [set()](http://www.cnblogs.com/mywood/p/7429135.html#set) | 创建一个无序不重复元素集，可进行关系测试，删除重复数据，还可以计算交集、差集、并集等。 | |
| [zip()](http://www.cnblogs.com/mywood/p/7429135.html#zip) | 用于将可迭代的对象作为参数，将对象中对应的元素打包成一个个元组，然后返回由这些元组组成的列表。 如果各个迭代器的元素个数不一致，则返回列表长度与最短的对象相同，利用 \* 号操作符，可以将元组解压为列表。 | | | | |
| [reversed()](http://www.cnblogs.com/mywood/p/7429135.html#reversed) | 用于反向排序列表中元素 | [enumerate()](http://www.cnblogs.com/mywood/p/7429135.html#enumerate) | 将一个可遍历的数据对象(如列表、元组或字符串)组合为一个索引序列，同时列出数据和数据下标，一般用在 for 循环当中 | | |

**创建函数：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [iter()](http://www.cnblogs.com/mywood/p/7429135.html#iter) | 用来生成迭代器。 | [range()](http://www.cnblogs.com/mywood/p/7429135.html#range) | 创建一个整数列表，一般用在 for 循环中。 | [dict()](http://www.cnblogs.com/mywood/p/7429135.html" \l "dict" \t "_blank) | 用于创建一个字典。 |
| [exec()](http://www.cnblogs.com/mywood/p/7429135.html#exec) | 执行储存在字符串或文件中的Python语句，相比于 eval，exec可以执行更复杂的 Python 代码 | | [bytearray()](http://www.cnblogs.com/mywood/p/7429135.html" \l "bytearray" \t "_blank) | 返回一个新字节数组。这个数组里的元素是可变的，并且每个元素的值范围: 0 <= x < 256。 | |
| [complex()](http://www.cnblogs.com/mywood/p/7429135.html#complex) | 创建一个值为 real + imag \* j 的复数或者转化一个字符串或数为复数。如果第一个参数为字符串，则不需要指定第二个参数。。 | | [xrange()](http://www.cnblogs.com/mywood/p/7429135.html" \l "xrange" \t "_blank) | 用法与 range 完全相同，所不同的是生成的不是一个数组，而是一个生成器。 | |

**获取函数：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [id()](http://www.cnblogs.com/mywood/p/7429135.html#id) | 用于获取对象的内存地址。 | [getattr()](http://www.cnblogs.com/mywood/p/7429135.html" \l "getattr" \t "_blank) | 返回一个对象属性值。 | [help()](http://www.cnblogs.com/mywood/p/7429135.html#help) | 用于查看函数或模块用途的详细说明。 |
| [next()](http://www.cnblogs.com/mywood/p/7429135.html#next) | 返回迭代器的下一个项目。 | [memoryview()](http://www.cnblogs.com/mywood/p/7429135.html" \l "memoryview" \t "_blank) | 返回给定参数的内存查看对象(Momory view)。 | [hash()](http://www.cnblogs.com/mywood/p/7429135.html#hash) | 用于获取取一个对象（字符串或者数值等）的哈希值。 |
| [vars()](http://www.cnblogs.com/mywood/p/7429135.html#vars) | 返回对象object的属性和属性值的字典对象。 | [globals()](http://www.cnblogs.com/mywood/p/7429135.html" \l "globals" \t "_blank) | 以字典类型返回当前位置的全部全局变量。 | [max()](http://www.cnblogs.com/mywood/p/7429135.html#max) | 返回给定参数的最大值，参数可以为序列。 |
| [property()](http://www.cnblogs.com/mywood/p/7429135.html#property) | 在新式类中返回属性值。 | [ord()](http://www.cnblogs.com/mywood/p/7429135.html" \l "ord" \t "_blank) | 返回一个字符对应的ASCII或者 Unicode值 | [staticmethod()](http://www.cnblogs.com/mywood/p/7429135.html" \l "staticmethod" \t "_blank) | 返回函数的静态方法 |
| [frozenset()](http://www.cnblogs.com/mywood/p/7429135.html#frozenset) | 返回一个不可变的集合，不能再添加或删除任何元素。 | [locals()](http://www.cnblogs.com/mywood/p/7429135.html#locals) | 以字典类型返回当前位置的全部局部变量。 | [unichr()](http://www.cnblogs.com/mywood/p/7429135.html" \l "unichr" \t "_blank) | 返回unicode字符 |
| [map()](http://www.cnblogs.com/mywood/p/7429135.html#map) | 根据提供的函数对指定序列做映射。 第一个参数 function 以参数序列中的每一个元素调用 function 函数，返回包含每次 function 函数返回值的新列表。 | | [dir()](http://www.cnblogs.com/mywood/p/7429135.html" \l "dir" \t "_blank) | 函数不带参数时，返回当前范围内的变量、方法和定义的类型列表；带参数时，返回参数的属性、方法列表。如果参数包含方法\_\_dir\_\_()，该方法将被调用。如果参数不包含\_\_dir\_\_()，该方法将最大限度地收集参数信息。 | |

**输入输出函数：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [input()](http://www.cnblogs.com/mywood/p/7429135.html#input) | 用来获取控制台输入，输入的值必须是一个合法的python表达式 | [print()](http://www.cnblogs.com/mywood/p/7429135.html#print) | 用于打印输出 | [open()](http://www.cnblogs.com/mywood/p/7429135.html#open) | 打开一个文件，创建一个file对象 |
| [file()](http://www.cnblogs.com/mywood/p/7429135.html#file) | 用于创建一个 file 对象，它有一个别名叫 open() | [raw\_input()](http://www.cnblogs.com/mywood/p/7429135.html" \l "raw_input" \t "_blank) | 接收控制台输入的内容，可接收所有的类型 |  |  |

**操作函数：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [delattr()](http://www.cnblogs.com/mywood/p/7429135.html#delattr) | 用于删除属性。 | [setattr()](http://www.cnblogs.com/mywood/p/7429135.html" \l "setattr" \t "_blank) | 对应函数 getatt()，用于设置属性值，该属性必须存在。 | [slice()](http://www.cnblogs.com/mywood/p/7429135.html#slice) | 切片分割对象，用在切片操作函数里的参数传递。 |
| [repr()](http://www.cnblogs.com/mywood/p/7429135.html#repr) | 将对象转化为供解释器读取的形式。 | [compile()](http://www.cnblogs.com/mywood/p/7429135.html#compile) | 将一个字符串编译为字节代码。 | [eval()](http://www.cnblogs.com/mywood/p/7429135.html" \l "eval" \t "_blank) | 执行一个字符串表达式，并返回表达式的值 |
| [execfile()](http://www.cnblogs.com/mywood/p/7429135.html#execfile) | 用来执行一个文件。 | [super()](http://www.cnblogs.com/mywood/p/7429135.html#super) | 调用下一个父类(超类)并返回该父类实例的方法 | [reload()](http://www.cnblogs.com/mywood/p/7429135.html#reload) | 重新载入之前载入的模块 |
| [type()](http://www.cnblogs.com/mywood/p/7429135.html#type) | 如果你只有第一个参数则返回对象的类型，三个参数返回新的类型对象。 | | [format()](http://www.cnblogs.com/mywood/p/7429135.html#format) | 格式化字符串的函数 str.format()，它增强了字符串格式化的功能。 | |
| [basestring()](http://www.cnblogs.com/mywood/p/7429135.html" \l "basestring" \t "_blank) | 是 str 和 unicode 的超类（父类），也是抽象类，因此不能被调用和实例化，但可以被用来判断一个对象是否为 str 或者 unicode 的实例 | | [classmethod()](http://www.cnblogs.com/mywood/p/7429135.html" \l "classmenthod" \t "_blank) | classmethod 修饰符对应的函数不需要实例化，不需要 self 参数，但第一个参数需要是表示自身类的 cls 参数，可以来调用类的属性，类的方法，实例化对象等。 | |
| [chr()](http://www.cnblogs.com/mywood/p/7429135.html" \l "chr" \t "_blank) | 用一个范围在 range（256）内的（就是0～255）整数作参数，返回一个对应的字符 | | [\_\_import\_\_()](http://www.cnblogs.com/mywood/p/7429135.html#__import__) | 动态加载类和函数 。 如果一个模块经常变化就可以使用 \_\_import\_\_() 来动态载入。 | |
| [reduce()](http://www.cnblogs.com/mywood/p/7429135.html#reduce) | 对参数序列中元素进行累积。 函数将一个数据集合（链表，元组等）中的所有数据进行下列操作：用传给reduce中的函数 function（有两个参数）先对集合中的第 1、2 个元素进行操作，得到的结果再与第三个数据用 function 函数运算，最后得到一个结果。 | | | | |

**判断函数：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [issubclass()](http://www.cnblogs.com/mywood/p/7429135.html" \l "issubclass" \t "_blank) | 判断参数 class 是否是类型参数 classinfo 的子类 | [hasattr()](http://www.cnblogs.com/mywood/p/7429135.html" \l "hasattr" \t "_blank) | 用于判断对象是否包含对应的属性。 | [isinstance()](http://www.cnblogs.com/mywood/p/7429135.html" \l "isinstance" \t "_blank) | 判断一个对象是否是一个已知的类型，类似 type()。 |
| [any()](http://www.cnblogs.com/mywood/p/7429135.html#any) | 用于判断给定的可迭代参数 iterable 是否全部为空对象，如果都为空、0、false，则返回 False，如果不都为空、0、false，则返回 True。 | | [all()](http://www.cnblogs.com/mywood/p/7429135.html#all) | 函数用于判断给定的可迭代参数 iterable 中的所有元素是否不为 0、''、False 或者 iterable 为空，如果是返回 True，否则返回 False | |
| [callable()](http://www.cnblogs.com/mywood/p/7429135.html#callable) | 检查一个对象是否是可调用的。如果返回True，object仍然可能调用失败；但如果返回False，调用对象ojbect绝对不会成功。 | | [cmp()](http://www.cnblogs.com/mywood/p/7429135.html" \l "cmp" \t "_blank) | 用于比较2个对象，如果 x < y 返回 -1, 如果 x == y 返回 0, 如果 x > y 返回 1。 | |

**下面是函数的详细介绍:**

**函数名：abs()**

**语法：**

abs( x )

**参数：数值表达式**

**返回值：函数返回x（数字）的绝对值。**

**代码示例：**

print ("abs(-45) : ", abs(-45))print ("abs(100.12) : ", abs(100.12))

**函数名：divmod()**

**语法：**

divmod(a, b)

**参数：a,b均为数值**

**返回值：一个包含商和余数的元组 例如（3，1），其中3是商，1是余数**

**代码示例：**

divmod(7, 2)

**函数名：pow()**

**语法：**

pow(x, y)

**参数：x,y均为数值**

**返回值：返回结果值，为数值类型**

**代码示例：**

pow(7, 2)

**函数名：sum()**

**语法：**

sum(a,b)

**参数：a为元素为数值的可迭代的对象， b为一个数值类型的参数，b可不写，不写时默认为0**

**返回值：返回值为迭代对象相加的和加上第二个参数，为数值类型**

**代码示例：**

a = sum((2, 3, 4), 1)

**函数名：bin()**

**语法：**

bin(x)

**参数：x为数值类型**

**返回值：返回值为二进制的表示，是字符串类型**

**代码示例：**

bin(12)

**函数名：oct()**

**语法：**

oct(x)

**参数：x为数值类型**

**返回值：返回值为八进制表示，是字符串类型**

**代码示例：**

oct(11)

**函数名：round()**

**语法：**

|  |  |
| --- | --- |
| 1 | round( m ,n  ) |

**参数：m为要四舍五入的值，为数值类型，n是保留的位数，为int类型**

**返回值：返回值为数值类型**

**代码示例：**

|  |  |
| --- | --- |
| 1 | round( 10.265 ,2  ) |

**函数名：long()**

**语法：**

|  |  |
| --- | --- |
| 1 | long(x, y) |

**参数：x为字符串或数值类型，b为进制数，默认10**

**返回值：结果为数值类型**

**代码示例：**

long(1)

**函数名：hex()**

**语法：**

hex(x)

**参数：x为十进制数值**

**返回值：十六进制的字符串**

**代码示例：**

hex(11)

**函数名：str()**

**语法：**

str(x)

**参数：x为对象**

**返回值：返回字符串**

**代码示例：**

str(object)

**函数名：int()**

**语法：**

int(x,y)

**参数：x为字符串类型**

**返回值：数值类型**

**代码示例：**

int("111111")

**函数名：tuple()**

**语法：**

tuple( seq )

**参数：seq为列表**

**返回值：返回元组**

**代码示例：**

tuple([1,2,3,4])

**函数名：bool()**

**语法：**

bool(x)

**参数：x为要转换的参数，可以是数值或者string**

**返回值：返回值为boolean类型，当参数为0或者空或者""的时候，返回false**

**代码示例：**

bool("")bool(0)

**函数名：float()**

**语法：**

float(x)

**参数：x为数值或者String**

**返回值：返回值为浮点数**

**代码示例：**

float("1")

**函数名：list()**

**语法：**

list( seq )

**参数：seq为元组**

**返回值：返回数组**

**代码示例：**

list(("12","456","789"))

**函数名：min()**

**语法：**

min( x, y, z, .... )

**参数：参数为序列或者集合**

**返回值：返回最小值**

**代码示例：**

min(("12","456","789"))

**函数名：sorted()**

**语法：**

sorted(iterable, cmp, key, reverse)

**参数：iterable为要排序的序列，cmp为比较函数，key为用来进行比较的元素，reverse为升序或者降序，默认为false降序**

**返回值：返回排好序的序列**

**代码示例：**

#对元组进行排序

sorted((12,123,56),reverse = True)

#利用cmp函数排序

sorted([('b',2),('a',1),('c',3),('d',4)], cmp=lambdax,y:cmp(x[1],y[1]))

#利用key排序

sorted([('b',2),('a',1),('c',3),('d',4)], key=lambda x:x[1])

**函数名：len()**

**语法：**

len( s )

**参数：s为字符、列表元组等**

**返回值：返回数值类型**

**代码示例：**

len("xtyz")

**函数名：filter()**

**语法：**

filter(function, iterable)

**参数：第一个参数为函数，函数返回布尔类型， 第二个参数为可迭代对象**

**返回值：返回过滤后的集合**

**代码示例：**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | def  tt (parm):      return parm < 7    a = filter(tt, [9,8,7,6,5,4,3])    print(list(a)) |

**函数名：set()**

**语法：**

|  |  |
| --- | --- |
| 1 | set(x) |

**参数：x为可迭代对象**

**返回值：返回一个不重复的元素集合**

**代码示例：**

复制代码

a = set("9876554321")print(a)#可以用来计算交集，并集，差集等x = set("987654")y = set("654321")print(x&y)#交集print(x|y)#并集print(x-y)#差集

复制代码

**函数名：zip()**

**语法：**

zip(a,b,c,d.....)

**参数：参数为多个可迭代对象**

**返回值：返回打包好的元组集合,元素合数与长度最短的迭代对象相同**

**代码示例：**

复制代码

x = "987654"

y = "654321"

print(list(zip(x,y)))

x = [1,2.3]

y = [4,5,6]

print(list(zip(x,y)))  
  
#解压  
z = zip(x,y)  
print(zip(\*z))

复制代码

**函数名：reverse()**

**语法：**

list.reverse()

**参数：没有参数**

**返回值：没有返回值**

**代码示例：**

x = (9,8,7,2,5,4,3)

x.reverse()

print(x)

#注意，是反向排列，不是按照大小排序

**函数名：enumerate()**

**语法：**

enumerate(sequence)

**参数：参数为可迭代对象**

**返回值：返回 enumerate(枚举) 对象**

**代码示例：**

seasons = ['Spring', 'Summer', 'Fall', 'Winter']

print(list(enumerate(seasons)))

**函数名：input()**

**语法：**

input()

**参数：参数无**

**返回值：返回输入值的类型（3 .x里 input() 默认接收到的是str 类型）**

**代码示例：**

**函数名：print()**

**语法：**

print(\*objects, sep=' ', end='\n', file=sys.stdout)

**参数：objects 为输出对象，可为多个，用，分隔； sep为间隔符，默认为" "; end为结尾的符号。默认是换行符"\n";file为要写入的文件对象**

**返回值：无**

**代码示例：**

print("111","222","333",sep ="===",end = "////////")

**函数名：open()**

**语法：**

open(name, mode, buffering)

**参数：name为要访问的文件或者创建的文件的字符串，可以包含路径，model为打开文件的模式，buffering : 缓冲区大小，设为0不会有寄存，设为1表示寄存行，大于1的整数，表示寄存去缓冲区的大小，**

**设置为负数，则为系统默认大小,文件打开模式如下**

复制代码

r #以只读方式打开文件。文件的指针将会放在文件的开头。这是默认模式。

rb #以二进制格式打开一个文件用于只读。文件指针将会放在文件的开头。这是默认模式。

r+ #打开一个文件用于读写。文件指针将会放在文件的开头。

rb+ #以二进制格式打开一个文件用于读写。文件指针将会放在文件的开头。

w #打开一个文件只用于写入。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。

wb #以二进制格式打开一个文件只用于写入。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。

w+ #打开一个文件用于读写。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。

wb+ #以二进制格式打开一个文件用于读写。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。

a #打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。

ab #以二进制格式打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。

a+ #打开一个文件用于读写。如果该文件已存在，文件指针将会放在文件的结尾。文件打开时会是追加模式。如果该文件不存在，创建新文件用于读写。

ab+ #以二进制格式打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。如果该文件不存在，创建新文件用于读写。

复制代码

**返回值：file对象**

**代码示例：**

f = open('test.txt')

**函数名：file()**

**语法：同open()**

**参数：同open()**

**返回值：同open()**

**代码示例：**

file('test.txt')

**函数名：raw\_input()**

**语法：**

raw\_input(x)

**参数：x为String，可不写，会在控制台输出x内容**

**返回值：返回输入的内容**

**代码示例：**

raw\_input\_

A = raw\_input("raw\_input:")

**函数名：iter()**

**语法：**

iter(x)

**参数：x为支持迭代的集合对象**

**返回值：迭代器对象**

**代码示例：**

lst = [1, 2, 3]for i in iter(lst):

 print(i)

**函数名：range()**

**语法：**

range(start, stop, step)

**参数：start为计数开始点，默认为0；stop为计数结束点，但不包括stop；step，步长，默认为1**

**返回值：列表对象**

**代码示例：**

range(10)range(1,5)range(1,10,2)

**函数名：dict()**

**语法：**

**参数：**

**返回值：字典对象**

**代码示例：**

a = dict() # 创建空字典

print(a)

a = dict(a='a', b='b', t='t') # 传入关键字

print(a)

a = dict(zip(['one', 'two', 'three'], [1, 2, 3])) # 映射函数方式来构造字典

print(a)a = dict([('one', 1), ('two', 2), ('three', 3)]) # 可迭代对象方式来构造字典

print(a)

**函数名：exec()**

**语法：**

exec x

**参数：x为可执行的字符串代码**

**返回值：无**

**代码示例：**

exec "print 'runoob.com'"

**函数名：bytearray()**

**语法：**

bytearray(source, encoding, errors)

**参数：source 可以是整数、字符串、可迭代对象**

**返回值：字节数数组**

**代码示例：**

bytearray([1,2,3])

**函数名：complex()**

**语法：**

complex(real, imag)

**参数：real 可以是 int, long, float或字符串，image可以是 int，long，float**

**返回值：返回一个复数**

**代码示例：**

complex(1, 2)

**函数名：xrange()**

**语法：**

xrange(start, stop, step)

**参数：start开始数，stop结束数，step步长**

**返回值：返回生成器**

**代码示例：**

xrange(8)

**函数名：id()**

**语法：**

id(object)

**参数：object为对象**

**返回值：对象的内存地址**

**代码示例：**

a = 'test'print(id(a))

**函数名：getattr()**

**语法：**

getattr(object, name, default)

**参数：object为对象，那么为属性名，default为默认值，如果属性不存在，返回默认值**

**返回值：属性值**

**代码示例：**

getattr(a, 'bar2', 3)

**函数名：help()**

**语法：**

help(x)

**参数：x为对象**

**返回值：返回对象的帮助信息**

**代码示例：**

help('sys') # 查看 sys 模块的帮助

**函数名：next()**

**语法：**

next(a, default)

**参数：a为可迭代对象，default为没有下一个元素时返回的值**

**返回值：可迭代对象的内容**

**代码示例：**

# 首先获得Iterator对象:it = iter([1, 2, 3, 4, 5])# 循环:while True: try: # 获得下一个值: x = next(it) print(x) except StopIteration: #遇到StopIteration就退出循环 break

**函数名：memoryview()**

**语法：**

memoryview(obj)

**参数：obj为对象**

**返回值：返回元组列表**

**代码示例：**

a = memoryview(b'babcefg')

print(a[1])

**函数名：hash()**

**语法：**

hash(x)

**参数：x 为对象**

**返回值: 返回对象的hash值**

**代码示例：**

a = hash('test')

print(a)

**函数名：vars()**

**语法：**

vars(obj)

**参数：obj为对象，如果没有的话，返回当前调用位置的属性和属性值**

**返回值：返回对象属性和属性值的字典对象**

class Runoob: a = 1print(vars(Runoob))

**代码示例：**

**函数名：globals()**

**语法：**

globals()

**参数：无**

**返回值：当前位置的全局变量**

**代码示例：**

print(globals())

**函数名：max()**

**语法**

max(a,b,x.....)

**参数：数值表达式**

**返回值：返回参数最大值**

**代码示例：**

print(max(80, 100, 1000))

**函数名：property()**

**语法：**

class property(fget, fset, fdel, doc)

**参数：**

* fget -- 获取属性值的函数
* fset -- 设置属性值的函数
* fdel -- 删除属性值函数
* doc -- 属性描述信息

**返回值：返回类属性**

**代码示例：**

**函数名：ord()**

**语法：**

ord(c)

**参数：c为字符**

**返回值：对应的十进制数**

**代码示例：**

ord('a')

**函数名：abs()**

**语法：**

abs( x )

**参数：x为数值表达式**

**返回值：x的绝对值**

**代码示例：**

abs( -8 )

**函数名：staticmethod()**

**语法：**

staticmethod(function)

**参数：无**

**返回值：无**

**代码示例：**

class C(object):

@staticmethod

def f():

print('runoob'); C.f(); # 静态方法无需实例化

cobj = C()cobj.f() # 也可以实例化后调用

**函数名：frozenset()**

**语法：**

frozenset([iterable])

**参数：可迭代对象**

**返回值：一个冻结的集合，不能修改或者删除**

**代码示例：**

frozenset(range(10))

**函数名：locals()**

**语法：**

locals()

**参数：无**

**返回值：返回当前位置的所有局部变量**

**代码示例：**

locals()

**函数名：unichr()**

**语法：**

|  |  |
| --- | --- |
| 1 | unichr(i) |

**参数：十进制或者十六进制的数字**

**返回值：unicode 的字符。**

**代码示例：**

unichr(97)

**函数名：map()**

**语法：**

map(function, iterable, ...)

**参数：**

* function -- 函数，有两个参数
* iterable -- 一个或多个序列

**返回值：返回列表**

**代码示例：**

def square(x) : # 计算平方数 return x \*\* 2 a = map(square, [1,2,3,4,5])

**函数名：dir()**

**语法：**

dir([object])

**参数：对象**

**返回值：返回属性列表**

**代码示例：**

**函数名：issubclass()**

**语法：**

issubclass(class, classinfo)

**参数：**

class--class类， classinfo---classifo类

**返回值：如果 class 是 classinfo 的子类返回 True，否则返回 False。**

**代码示例：**

class A: passclass B(A): pass print(issubclass(B,A)) # 返回 True

**函数名：hasattr()**

**语法：**

hasattr(object, name)

**参数：object为对象，那么为属性名**

**返回值：如果对象内有这个属性，返回true，否则false、**

**代码示例：**

复制代码

class Coordinate:

x = 10

y = -5

z = 0

point1 = Coordinate()

print(hasattr(point1, 'x'))

print(hasattr(point1, 'y'))

print(hasattr(point1, 'z'))

print(hasattr(point1, 'no')) # 没有该属性

**函数名：isinstance()**

**语法：**

isinstance(object, classinfo)

**参数：**

* + object -- 实例对象。
  + classinfo -- 可以是直接或间接类名、基本类型或者有它们

**返回值：如果对象的类型与参数二的类型（classinfo）相同则返回 True，否则返回 False。。**

**代码示例：**

class A: pass class B(A): pass isinstance(A(), A) # returns Truetype(A()) == A # returns Trueisinstance(B(), A) # returns Truetype(B()) == A # returns False

**函数名：any()**

**语法：**

any(iterable)

**参数：元组或者列表**

**返回值：如果都为空、0、false，则返回false，如果不都为空、0、false，则返回true。**

**代码示例：**

|  |  |
| --- | --- |
| 1 | any(['a', 'b', 'c', 'd'])  # 列表list，元素都不为空或0 |

**函数名：all()**

**语法：**

all(iterable)

**参数：元组或列表**

**返回值：如果iterable的所有元素不为0、''、False或者iterable为空，all(iterable)返回True，否则返回False；**

**代码示例：**

all(['a', 'b', 'c', 'd']) # 列表list，元素都不为空或0

**函数名：callable()**

**语法：**

callable(object)

**参数：对象**

**返回值：可调用返回 True，否则返回 False。**

**代码示例：callable(0)**

**函数名：cmp()**

**语法：**

**参数：**

**返回值：**

**代码示例：**

**函数名：delattr()**

**语法：**

cmp( x, y )

**参数：都是数值表达式**

**返回值：如果 x < y 返回 -1, 如果 x == y 返回 0, 如果 x > y 返回 1。**

**代码示例：**

cmp(80, 100)

**函数名：setattr()**

**语法：**

setattr(object, name, value)

**参数：**

* object -- 对象。
* name -- 字符串，对象属性。
* value -- 属性值。

**返回值：无**

**代码示例：**

class A(object): bar = 1 a = A()getattr(a, 'bar') # 获取属性 bar 值setattr(a, 'bar', 5) # 设置属性 bar 值a.bar

**函数名：slice()**

**语法：**

slice(start, stop, step)

**参数：**

* start -- 起始位置
* stop -- 结束位置
* step -- 间距

**返回值：一个切片对象**

**代码示例：**

slice(5)

**函数名：repr()**

**语法：**

repr(object)

**参数：对象**

**返回值：对象的String格式**

**代码示例：**

s = 'RUNOOB'repr(s)

**函数名：compile()**

**语法：**

compile(source, filename, mode, flags, dont\_inherit)

**参数：**

* source -- 字符串或者AST（Abstract Syntax Trees）对象。。
* filename -- 代码文件名称，如果不是从文件读取代码则传递一些可辨认的值。
* mode -- 指定编译代码的种类。可以指定为 exec, eval, single。
* flags -- 变量作用域，局部命名空间，如果被提供，可以是任何映射对象。。
* flags和dont\_inherit是用来控制编译源码时的标志

**返回值：**

**代码示例：**

**函数名：eval()**

**语法：**

eval(expression, globals, locals)

**参数：**

* expression -- 表达式。
* globals -- 变量作用域，全局命名空间，如果被提供，则必须是一个字典对象。
* locals -- 变量作用域，局部命名空间，如果被提供，可以是任何映射对象。

**返回值：表达式执行结果**

**代码示例：**

eval( '3 \* x' )

**函数名：execfile()**

**语法：**

execfile(filename, globals, locals)

**参数：**

* filename -- 文件名。
* globals -- 变量作用域，全局命名空间，如果被提供，则必须是一个字典对象。
* locals -- 变量作用域，局部命名空间，如果被提供，可以是任何映射对象。

**返回值：执行结果**

**代码示例：**

execfile('hello.py') #需要有hello.py

**函数名：super()**

**语法：**

super(type[, object-or-type])

**参数：**

* type -- 类。
* object-or-type -- 类，一般是 self

**返回值：无**

**代码示例：**

class FooParent(object):

def \_\_init\_\_(self):

self.parent = 'I\'m the parent.'

print ('Parent')

def bar(self,message):

print ("%s from Parent" % message)

class FooChild(FooParent):

def \_\_init\_\_(self):

# super(FooChild,self) 首先找到 FooChild 的父类（就是类 FooParent），然后把类B的对象 FooChild 转换为类 FooParent 的对象

 super(FooChild,self).\_\_init\_\_()

print ('Child')

defbar(self,message):

super(FooChild, self).bar(message)

 print ('Child bar fuction')

 print (self.parent)

 if \_\_name\_\_ == '\_\_main\_\_':

fooChild = FooChild()

fooChild.bar('HelloWorld')

**函数名：reload()**

**语法：**

reload(module)

**参数：模块对象**

**返回值：返回模块对象**

**代码示例：**

reload(sys)

**函数名：type()**

**语法：**

type(name, bases, dict)

**参数：**

* name -- 类的名称。
* bases -- 基类的元组。
* dict -- 字典，类内定义的命名空间变量。

**返回值：一个参数返回对象类型, 三个参数，返回新的类型对象。**

**代码示例：**

type(1)type('runoob')type([2])type({0:'zero'})x = 1 type( x ) == int # 判断类型是否相等print(x)# 三个参数class X(object): a = 1X = type('X', (object,), dict(a=1)) # 产生一个新的类型 Xprint(X)

**函数名：format()**

**语法：String.format()**

**参数：**

**返回值：格式化之后的字符串**

**代码示例：**

"{} {}".format("hello", "world") # 不设置指定位置，按默认顺序"{0} {1}".format("hello", "world") # 设置指定位置"{1} {0} {1}".format("hello", "world") # 设置指定位置# 通过字典设置参数site = {"name": "菜鸟教程", "url": "www.runoob.com"}print("网站名：{name}, 地址 {url}".format(\*\*site))my\_list = ['菜鸟教程', 'www.runoob.com']print("网站名：{0[0]}, 地址 {0[1]}".format(my\_list)) # "0" 是可选的class AssignValue(object): def \_\_init\_\_(self, value): self.value = valuemy\_value = AssignValue(6)print('value 为: {0.value}'.format(my\_value)) # "0" 是可选的

**函数名：basestring()**

**语法：**

basestring()

**参数：无**

**返回值：无**

**代码示例：**

isinstance("Hello world", basestring)#判断一个对象是否为 str 或者 unicode 的实例

**函数名：classmethod()**

**语法：**

classmethod

**参数：无**

**返回值：返回函数的类方法**

**代码示例：**

复制代码

class A(object): bar = 1 def func1(self): print ('foo') @classmethod def func2(cls): print ('func2') print (cls.bar) cls().func1() # 调用 foo 方法 A.func2() # 不需要实例化

复制代码

**函数名：chr()**

**语法：**

chr(i)

**参数：十进制或者十六进制的数字,0-255之间**

**返回值：对应的ASCLL字符**

**代码示例：**

chr(123)

**函数名：\_\_import\_\_()**

**语法：**

\_\_import\_\_(name)

**参数：模块名**

**返回值：元组数组**

**代码示例：**

import sys \_\_import\_\_('a') # 导入 a.py 模块

**函数名：reduce()**

**语法：**

reduce(function, iterable, initializer)

**参数：**

* function -- 函数，有两个参数
* iterable -- 可迭代对象
* initializer -- 可选，初始参数

**返回值：返回计算结果**

**代码示例：**

def add(x, y) : # 两数相加 return x + y reduce(add, [1,2,3,4,5]) # 计算列表和：1+2+3+4+5reduce(lambda x, y: x+y, [1,2,3,4,5]) #使用 lambda 匿名函数