03: 常见内置函数

abs(x)

返回一个数的绝对值。参数可以是普通的整数，长整数或者浮点数。如果参数是个复数，返回它的模。

>>> abs(1)

1

>>> abs(-1)

1

divmod(x,y)

结合除法和余数运算，返回包含商和余数的元组

>>> divmod(5,2)

(2, 1)

pow(x, y[, z])

pow(x,y)返回x的y次方，如果z存在即pow(x,y,z)，其结果等于pow(x,y)%z

>>> pow(2,3)

8

>>> pow(2,3,3)

2

>>> pow(2,-3)

0.125

round(x[, n])

返回浮点数x的四舍五入的值，其中n值，表示小数点后的位数。

>>> round(2)

2.0

>>> round(2.567)

3.0

>>> round(2.567, 2)

2.57

min(x[,y,z…]), max(x[,y,z…])

返回给定参数和最小值或者最大值，参数可以为序列。

>>> min(2,1)

1

>>> max(2,1)

2

sum(iterable[,start])

sum()函数对序列进行求和运算

>>> sum([1,2,3,4])

10

>>> sum(range(101))

5050

oct(x)

oct()函数用于将一个整数转换为八进制

>>> oct(0)

'0'

>>> oct(4)

'04'

>>> oct(8)

'010'

>>> oct(10)

'012'

hex(x)

hex()函数用于将10进制数转换为16进制数

>>> hex(255)

'0xff'

>>> hex(256)

'0x100'

>>> hex(16)

'0x10'

chr(i)

char()函数根据输入的数字返回一个字符，数字的范围为range(0,256)

>>> chr(65)

'A'

bin(x)

bin()函数返回一个整数的二进制表示。

>>> bin(1)

'0b1'

>>> bin(10)

'0b1010'

类型转换

bool([x])

bool()函数将参数转换为bool类型，如果没有给参数则返回False

>>> bool()

False

>>> bool(0)

False

>>> bool(1)

True

int(x,base=10)

int()函数用于将一个字符串或者数字转换为整型。

参数base表示进制数，默认为十进制

>>> int()

0

>>> int(1.1)

1

>>> int('10',16)

16

>>> int('10',8)

8

float([x])

float()函数用于将整数和字符串转换成浮点数

>>> float()

0.0

>>> float(1)

1.0

str(object)

str()函数将对象对象转换为可阅读的样式。

>>> str()

''

>>> str(str)

"<type 'str'>"

>>> str('1234')

'1234'

list([iterable])

list()函数根据输入的可迭代的参数，创建一个新的列表。

>>> list()

[]

>>> list('1234')

['1', '2', '3', '4']

tuple([iterable])

tuple()函数根据输入的可迭代的参数，创建一个新的元组。

>>> tuple()

[]

>>> tuple('1234')

['1', '2', '3', '4']

set([iterable])

set()函数根据参数创建一个新的集合

>>> set()

set([])

>>> set('1234')

set(['1', '3', '2', '4'])

frozenset([iterable])

frozenset()函数返回一个冻结的set集合，其不能添加和删除元素

>>> frozenset()

frozenset([])

>>> frozenset(range(10))

frozenset([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

序列相关

len(object)

返回字符串或者序列的长度

>>> len("hello world!")

12

>>> len([1,2,3,4,5])

5

range([start,]stop[,step])

创建一个整数列表，一般用在for循环中。

其中的参数说明为：

start : 生成的整数从start开始，可省略，默认为0.

end : 生成的整数以end结束，但不包含end。

step : 步长，默认为1

>>> range(5)#等于range(0,5)

[0, 1, 2, 3, 4]

>>> range(1, 5)#不包含5

[1, 2, 3, 4]

>>> range(1,5,2)# 步长为2

[1, 3]

>>> range(0)# range(0,0)为空

[]

>>> range[5,1] # start不能小于end

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'builtin\_function\_or\_method' object has no attribute '\_\_getitem\_\_'

xrange([start,]stop[,step])

xrange和range完全相同，使用方法相同，但是xrange生成的不是一个数组，而是一个生成器

，其只有在需要时才回生成列表值，当列表很大的时候，可以节省内存。

zip([iterable,…])

zip()函数聚合传入的一个到多个迭代器相同位置的元素生成一个新的元组类型的迭代器

>>> x = [1,2,3]

>>> y = [a,b,c]

>>> z = [4,5,6]

>>> zip(x,y)

[(1, 'a'), (2, 'b'), (3, 'c')]

>>> zip(x,y,z)

[(1, 'a', 4), (2, 'b', 5), (3, 'c', 6)]

sorted()

sorted()函数对所有可迭代的对象进行排序操作。

其原型为：sorted(iterable, cmp=None, key=None, reverse=False)

其主要参数含义为

cmp: 比较函数。

key: 主要用来比较的元素

reverse: 排序方式：True:降序排列；False:升序排列

>>> sorted([4,3,2,1]) # 默认为升序排列

[1, 2, 3, 4]

>>> sorted([1,2,3,4], reverse=False) #升序排列

[1, 2, 3, 4]

>>> sorted([1,2,3,4], reverse=True) #降序排列

[4, 3, 2, 1]

reversed(seq)

reversed()反转序列，生成一个新的序列

>>> list(reversed([1,2,3,4]))

[4, 3, 2, 1]

enumerate(iterable[, start])

enumerate()函数用于将一个可遍历的数据对象，组合为一个索引序列，同时包含索引和数据。

其中参数start表示参数下标的起始位置。

>>> a = [1,2,3,4]

>>> list(enumerate(a))

[(0, 1), (1, 2), (2, 3), (3, 4)]

>>> list(enumerate(a,2))

[(2, 1), (3, 2), (4, 3), (5, 4)]

iter

iter()函数用于创建一个迭代器

>>> a = iter(range(10))

>>> next(a)

0

>>> next(a)

1

>>> next(a)

2

next()

next()函数返回迭代器的下一个元素

>>> a = iter(range(10))

>>> next(a)

0

>>> next(a)

1

>>> next(a)

2

slice

slice()函数返回一个切片对象，主要用在切片操作函数里的参数传递。

slice0\_5 = slice(5)#返回0-5切片的数据

slice4\_10 = slice(4,10)#返回4-10切片的数据

print range(10)[slice0\_5]#[0, 1, 2, 3, 4]

print range(10)[slice4\_10]#[4, 5, 6, 7, 8, 9]

all(iterable)

all()函数用于判断给定的可迭代参数中是否含有元素为0、’’、False，如果含有这样的元素

返回False否则返回True，如果是空的迭代器则返回True。

>>> all([])#空迭代器

True

>>> all([1,2,3,4])#所有元素都不为0、''、False

True

>>> all([0,1,2,3,4])#含有元素 0

False

>>> all(['',1,2,3,4])# 含有元素 ''

False

>>> all([False,1,2,3,4])# 含有元素 False

False

any(iterable)

any()函数用于判断给定的迭代器是否全部为空的元素0、’’、False,

如果全部为空则返回False，否则返回True。

空的元组或者列表返回False.

>>> any([]) #空列表返回False

False

>>> any([1]) #没有空元素

True

>>> any([0,1]) #存在非空元素

True

>>> any([0]) #不存在非空元素

False

all()和any()的函数的区别如下：

all()函数只要含有空元素，那么就返回False。

any()函数只要含有非空元素，那么就返回True。

空列表或者元祖，all()函数返回true，any()函数返回False

cmp(x,y)

cmp()函数用于比较两个对象的大小，如果x<y返回-1，如果相等则返回0，如果x>y则返回1.

>>> cmp(1,2)

-1

>>> cmp(1,1)

0

>>> cmp(2,1)

filter(function or None, sequence)

filter()函数用于过滤序列，过滤不符合的数据，返回有符合元素组成的序列。

#-\*-coding:utf-8 -\*-

#过滤大于5的数

def max5(s) :

return s > 5

print filter(max5, range(10))