第17课 了解其它常用核心函数2

序列操作类：all、any、filter、map

all() 函数用于判断给定的可迭代参数中的所有元素是否都为 TRUE，如果是返回 True，否则返回 False。元素除了是 0、空、None、False 外都算 True。

以下展示了使用 all() 方法的实例：

序列操作类：all、any、filter、map

```
>>> all(['a', 'b', 'c', 'd'])  # 列表list，元素都不为空或0
True
>>> all(['a', 'b', '', 'd'])   # 列表list，存在一个为空的元素
False
>>> all([0, 1, 2, 3])          # 列表list，存在一个为0的元素
False

>>> all(('a', 'b', 'c', 'd'))  # 元组tuple，元素都不为空或0
True
>>> all(('a', 'b', '', 'd'))   # 元组tuple，存在一个为空的元素
False
>>> all((0, 1, 2, 3))          # 元组tuple，存在一个为0的元素
False
>>> all([])             # 空列表
True
>>> all(())             # 空元组
True
```

序列操作类：all、any、filter、map

any() 函数用于判断给定的可迭代参数 iterable 是否全部为 False，则返回 False，如果有一个为 True，则返回 True。

元素除了是 0、空、FALSE 外都算 TRUE。

序列操作类：all、any、filter、map

```
>>>any(['a', 'b', 'c', 'd'])  # 列表list，元素都不为空或0
True
>>> any(['a', 'b', '', 'd'])   # 列表list，存在一个为空的元素
True
>>> any([0, '', False])        # 列表list,元素全为0,'',false
False
>>> any(('a', 'b', 'c', 'd'))  # 元组tuple，元素都不为空或0
True
>>> any(('a', 'b', '', 'd'))   # 元组tuple，存在一个为空的元素
True
>>> any((0, '', False))        # 元组tuple，元素全为0,'',false
False
>>> any([]) # 空列表
False
>>> any(()) # 空元组
False
```

序列操作类：all、any、filter、map

filter() 函数用于过滤序列，过滤掉不符合条件的元素，返回一个迭代器对象，如果要转换为列表，可以使用 list() 来转换。

该接收两个参数，第一个为函数，第二个为序列，序列的每个元素作为参数传递给函数进行判断，然后返回 True 或 False，最后将返回 True 的元素放到新列表中。

序列操作类：all、any、filter、map

filter 函数的实例：过滤出列表中的所有奇数：

```
def jishu(n):
    return n % 2 == 1

newlist = filter(jishu, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
n=list(newlist)
print(n)
```

序列操作类：all、any、filter、map

map是python内置函数，会根据提供的函数对指定的序列做映射。
map()函数的格式是：map(function,iterable,...)
第一个参数接受一个函数名，后面的参数接受一个或多个可迭代的序列。

序列操作类：all、any、filter、map

把函数依次作用在list中的每一个元素上，得到一个新的list并返回。注意，map不改变原list，而是返回一个新list。

```
del square(x):
    return x ** 2

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

# 结果如下:
[1,4,9,16,25]
```

序列操作类：all、any、filter、map

通过使用lambda匿名函数的方法使用map()函数：

```
map(lambda x, y: x+y,[1,3,5,7,9],[2,4,6,8,10])

# 结果如下：
[3,7,11,15,19]
```

序列操作类：all、any、filter、map

通过lambda函数使返回值是一个元组：

```
map(lambdax, y : (x**y,x+y),[2,4,6],[3,2,1])

# 结果如下
[(8,5),(16,6),(6,7)]
```

序列操作类：all、any、filter、map

当不传入function时，map()就等同于zip()，将多个列表相同位置的元素归并到一个元组：

```
map(None,[2,4,6],[3,2,1])

# 结果如下
[(2,3),(4,2),(6,1)]
```

序列操作类：all、any、filter、map

通过map还可以实现类型转换

将元组转换为list：

```
map(int,(1,2,3))

# 结果如下：
[1,2,3]
```

序列操作类：all、any、filter、map

通过map还可以实现类型转换

将字符串转换为list：

```
map(int,'1234')

# 结果如下：
[1,2,3,4]
```

序列操作类：all、any、filter、map

通过map还可以实现类型转换

提取字典中的key，并将结果放在一个list中：

```
map(int,{1:2,2:3,3:4})

# 结果如下
[1,2,3]
```

对象操作类：help、dir、ascii、vars

help() 函数用于查看函数或模块用途的详细说明。返回对象帮助信息。

以下实例展示了 help 的使用方法：

```
>>>help('sys')                    # 查看 sys 模块的帮助
……显示帮助信息……
>>>help('str')                    # 查看 str 数据类型的帮助
……显示帮助信息……
>>>a = [1,2,3]
>>>help(a)                        # 查看列表 list 帮助信息
……显示帮助信息……
>>>help(a.append)                 # 显示list的append方法的帮助
……显示帮助信息……
```

对象操作类：help、dir、ascii、vars

dir() 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。

dir([object])，参数说明：object -- 对象、变量、类型。

返回模块的属性列表。

例如：

```
>>>dir()    #   获得当前模块的属性列表
>>> dir([ ])     # 查看列表的方法
```

对象操作类：help、dir、ascii、vars

ascii()函数返回一个表示对象的字符串，但是对于字符串中的非 ASCII 字符则返回通过 repr() 函数使用 \x, \u 或 \U 编码的字符，生成字符串。

对象操作类：help、dir、ascii、vars

```
>>> ascii((1,2))
'(1, 2)'
>>> ascii([1,2])
'[1, 2]'
>>> ascii({1:2,'name':5})
"{1: 2, 'name': 5}"
>>> ascii('?')
"'?'"
>>> ascii(set([1,1,2,3]))
'{1, 2, 3}'
>>>
```

对象操作类：help、dir、ascii、vars

vars() 函数返回对象object的属性和属性值的字典对象。

vars() 函数语法：vars([object]。

返回对象object的属性和属性值的字典对象，如果没有参数，就打印当前调用位置的属性和属性值

例如：

```
>>> vars(int)
```