6.6 内置函数

Built-in Functions				
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

sorted函数

- sorted函数对字符串，列表，元组，字典等对象进行排序操作。
- sort是应用在list上的方法，sorted可以对更多的数据类型进行排序操作。
- 即便都是对列表操作，list的sort方法返回的是对已经存在的列表进行操作，而内建函数sorted返回的是一个新的list，而不是在原来的基础上进行的操作。
-

sorted函数语法

- sorted(iterable[,key[, reverse]])
- iterable -- 序列，如字符串，列表，元组等。
- key – 函数，缺省为空
- reverse -- 排序规则
- reverse = True 降序， reverse = False 升序（默认）。

表格排序

姓名	分数	年龄
江幸	89	15
方鹏	80	14
陈可	85	14

- >>>students = [('江幸',89, 15), (' 方鹏',80, 14), ('陈可', 85, 14)]
- #第二个分量是成绩，第三个分量是年龄
-
- >>>print(sorted(students, key=lambda s: s[2])) # 按年龄从小到大排序
- [(' 方鹏', 80, 14), ('陈可', 85, 14), ('江幸', 89, 15)]
-
-
- >>>print(sorted(students, key=lambda s: s[1], reverse=True))# 按成绩从大到小降序
- [('江幸', 89, 15), ('陈可', 85, 14), (' 方鹏', 80, 14)]

map函数

- map会根据提供的函数对指定序列做映射。
- map函数语法：map(function, iterable, ...)
- 第一个参数 function是对参数序列中的每一个元素调用 function 函数，iterable是序列，
- 返回值的新列表或迭代器，每个元素是调用
- function 函数的返回值

map函数举例

- >>>print(list(map(lambda x: x ** 2, [1, 2, 3, 4, 5]))) # 使用 lambda 匿名函数
- [1, 4, 9, 16, 25]
-
-
- >>>print(list(map(lambda x, y: x + y, [1, 3, 5, 7, 9], [2, 4, 6, 8, 10])))
- # 提供了两个列表，对相同位置的列表数据进行相加
- [3, 7, 11, 15, 19]

zip函数

- zip() 函数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或迭代器 。
- 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。
- zip 语法：zip([iterable, ...])

  参数说明：iterable,… --两个或多个序列

  返回值：返回元组列表

zip函数举例

- >>>a = [1,2,3]
- >>>b = [4,5,6]
- >>>c = [4,5,6,7,8]

- print(list(zip(a,b)))
- [(1, 4), (2, 5), (3, 6)]
- print(list(zip(a,c))) # 元素个数与最短的列表一致
- [(1, 4), (2, 5), (3, 6)]

字典键值互换

- >>>d={‘blue’:500,‘red’:100,‘white’:300} #值不相同
- >>>d1=dict(zip(d.values(),d.keys())
- >>>print(d1)
- {500: 'blue', 100: 'red', 300: 'white'}

eval和exec函数

- Python是一种动态语言，它包含很多含义。Python变量类型，操作的合法性检查都在动态运行中检查；运算的代码需要到运行时才能动态确定；程序结构也可以动态变化，容许动态加载新模块等。这两个函数就体现了这个特点。
- eval是计算表达式,返回表达式的值。
- **e**xec可运行Python的程序,返回程序运行结果。

eval和exec函数举例

- >>>x,y=3,7
- >>>eval(‘x+3*y-4’)
- **20**

- >>>exec(‘print(“hello world”)’)
- Hello world

all和any函数

- all()和any()函数将可迭代的对象作为参数。
- all函数参数都是True时，才返回True，否则返回False
- any函数参数只要有一个为True，就返回True,参数全部是False时，返回False.

all和any函数举例

- >>> n=47
- >>> all([1 if n%k!=0 else 0 for k in range(2,n)])
- True
- 思考：True背后的含义？
- >>> n=15
- >>> all([1 if n%k!=0 else 0 for k in range(2,n)])
- False
- >>> any([[],False,0])　#空列表和0都表示False
- False