# 序列：字符串、列表和元组

## 序列

序列包含字符串、列表和元组，所以其内容也使用它们

### 1、标准类型操作符

1.1、对象值比较 <、>、<=、>=、==、!=

判断对象值是否相等，不一定是同一个对象，返回True/False

数值比较大小和符号，字符串按序列值比较

1.2、对象身份比较 is/is not 等价于 id(obj1)==id(obj2)

判断是否指向同一个对象

1.3、布尔类型 not/and/or

## 附：

1、标准类内建函数

str(obj)、isinstance(子,父)

2、类型工厂函数

**以前**所谓的内建转换函数像int()、type()、list()等，现在都成了工厂函数

即是说，它们看上去像函数，实际上是类，当调用时，生成该类型的一个实例，就像工厂生产货物一样

3、python方法中有可变参数、关键字参数，因此不支持方法重载，只有重写（继承）

### 2、序列类型操作符

2.1、成员关系操作符 in/not in

2.2、连接操作符 +

对字符串来说，不如将子字符串放到一个列表或可迭代对象中，然后用join()方法拼接节约内存

对列表来说，推荐用列表的extend()方法将多个列表对象合并

2.3、重复操作符 \*

同连接操作符一样，返回一个新的包含多份原对象拷贝的对象

2.4、切片操作符 []/[:]/[::] 切片是**前闭后开**

sequence/sequence[:]

sequence[0:3]/sequence[:3]

sequence[2:5]/sequence[2:]

sequence[3]

sequence[::-1]

正索引以开始为起点，负索引以结束为起点

### 3、内建函数（BIF）

3.1、类型转换

各序列类型之间转换list()、str()、tuple()等是工厂函数，将序列对象做参数，并将其内容（浅）拷贝到新生成的对象中

3.2、序列类型可用的内建函数

enumerate(iter) 返回索引index和值item组成的元组

len(seq) 返回长度

max() 返回最大值

min() 返回最小值

reversed(seq) 参数为序列，返回逆序的迭代器

sorted(seq) 返回序列

sum()

zip(it0, it1,.....) 返回列表，里面包含元组，各序列的的第一个元素组成第一个元组，.......

\*zip() 拆包，同zip相反，参数为列表，里面包含元组，返回列表

## 字符串

字符串不变性，是不可变对象

字符串建议用join()拼接，参数为列表或元组

格式化操作符 %

字符串类型函数

工厂函数 str()

chr() 参数为0~255，返回单字符

ord() 参数为单字符，返回对应ASCII数值

unichr() 参数为单字符，返回对应Unicode数值

字符串内建函数

string.decode(encoding=’UTF-8’)

string.encode(encoding=’UTF-8’)

string.startswith(obj,beg=0,end=len(string))

string.endswith(obj,beg=0,end=len(string))

string.upper()/lower() 大/小写

string.join(seq)

string.split(str)/splitlines() 以 str/行 分割

string.strip()/lstrip()/rstrip()

## 列表

dir()方法开得到一个对象（万物皆对象）的所以方法和属性

如dir(list)/dir([])

列表类型的内建函数

list.append(obj) 添加

list.count(obj) 返回出现次数

list.extend(seq) 将序列的内容加入列表中

list.index(obj) 返回索引值

list.insert(index,obj) 在index处插入obj

list.pop(index=-1) 删除并返回删掉的值，默认最后一个

list.remove(obj) 删除obj

list.reverse() 翻转列表

list.sort()

注意：方法和函数

注意：可变对象的方法是没有返回值（可变对象本身）的

不可变对象是不能改变值的，所以要返回一个新对象

列表可以用来构建其他数据结构如：堆栈（先进后出）和队列（先进先出）

## 元组

元组是不可变类型，因此，可做一些列表不可做的事如用作字典的key

工厂方法tuple(1,) 逗号

不可变类型，没有‘更新’

元组本身不可变，但其元素未必，如元组中包含列表

## 附：

对一个对象进行浅拷贝就是新建一个类型跟原对象一样，其内容是原对象元素的引用，即这个拷贝的对象本身是新的，但它得到内容不是

非容器类型（数字、字符串、...）没有被拷贝一说

元组变量包含原子类型对象，深拷贝不会进行

copy模块只有copy()和deepcopy()两个函数