1. 列表

标志：[]

基本操作(创建，append( )，pop( ) ,del( ), 拷贝）

append() 是将一个对象作为一个整体添加到列表中，追加到最后，整体追加

extend() 追加最后，拆分追加

insert() 指定插入位置，整体插入

**pop()** 出栈操作，遵循 先入后出 原则，append()代替push()作为入栈操作

del member[索引值]

深拷贝 deep copy，深拷贝通常只对可变对象进行深层复制，不可变对象通常不会被复制，例如：import copy

L[3.1,3.2]

L1=[1,2,L]

L2=copy.deepcopy(L1) 深拷贝 递归地拷贝

L[0]=3.14

L1 =[1,2,[3.14,3.2]]

L2=[1,2,[3.1,3.2]] 此列表不受影响

浅拷贝 shallow copy，是指在复制过程当中，只会复制一层变量，不会复制深层变量，例如：L=[3.1,3.2]

L1=[1,2,L]

L2=L1.copy() 浅拷贝

L[0]=3.14

L1 =[1,2,[3.14,3.2]]

L2=[1,2,[3.14,3.2]]

L1[2] 和 L2[2] 共同拥有L绑定的对象

python中的变量不存储任何数据，只是一种绑定关系，不占用内存，运行快，效率高

列表相关方法

L.index(v [, begin[, end]])

返回对应元素的索引下标, begin为开始索引，end为结束索引,当 value 不存在时触发ValueError错误

L.insert(index, obj)

将某个元素插放到列表中指定的位置

L.count(x)

返回列表中元素的个数

L.remove(x)

从列表中删除第一次出现在列表中的值

L.copy()

复制此列表（只复制一层，不会复制深层对象)

L.append(x)

向列表中追加单个元素

L.extend(lst)

向列表追加另一个列表

L.clear()

清空列表,等同于 L[:] = []

L.sort(reverse=False)

将列表中的元素进行排序，默认顺序按值的小到大的顺序排列

L.reverse()

列表的反转，用来改变原列表的先后顺序

L.pop([index])

删除索引对应的元素，如果不加索引，默认删除最后元素，同时返回删除元素的引用关系

2.元组

标志：()

基本操作（创建及不可变性）

元组与多数python集合（列表、字典、集，等等）一样，保存的是对象的引用。如果引用的元素是可变的，即便元组本身不可变，但是元素依然可变。也就是说元组的不可变性其实是指tuple数据结构的物理内容（即保存的引用）不可变，与引用的对象无关。

3.string字符串

定义及基本操作（+，\*，读取方式）

字符串是 Python 中最常用的数据类型。我们可以使用引号('，"，''')来创建字符串。

创建字符串很简单，只要为变量分配一个值即可。它是一种不可变的数据类型。

+ 字符串连接

\* 重复输出字符串

[] 通过索引获取字符串中字符

[ : ] 截取字符串中的一部分

In 成员运算符 - 如果字符串中包含给定的字符返回 True

not in 成员运算符 - 如果字符串中不包含给定的字符返回 True

字符串相关方法

len（）：求出给定字符串的长度；

count（）：求出特定字符出现的次数

find（）：找出具体的字符出现的位置，返回第一次出现的位置

* repr（）：可以将不可见字符显示出来
* replace（）：replace（“需要替换的字符”，“替换后的字符”，替换的次数）
* split（）：把字符串以某个字符进行切割，默认为不可见的字符，切割后为列表
* join（）：格式为：“以什么字符连接” .  join ( 待连接的字符串 )

4.字符串格式化问题

在Python3引入了一个新的字符串格式化的方法，并且随后支持了Python2.7。这个“新式”的字符串格式化方法摆脱了%操作符并且使得字符串格式化的语法更规范了。 现在时候通过调用字符串对象的.format() 方法进行格式化。基本语法是通过 {} 和 : 来代替以前的 %例如：

'Hello, {}'.format(name)

可以通过变量名来进行替换，且不用担心变量的位置。这个强大的特性可以在不改变format()的传入参数的情况下，允许重新调整显示的位置，如：

'Hey {name}, there is a 0x{errno:x} error!'.format(     name=name, errno=errno)

>>>"{} {}".format("hello", "world") # 不设置指定位置，按默认顺序 'hello world' >>> "{0} {1}".format("hello", "world") # 设置指定位置 'hello world' >>> "{1} {0} {1}".format("hello", "world") # 设置指定位置 'world hello world'

也可以设置参数：

print("网站名：{name}, 地址 {url}".format(name="菜鸟教程", url="www.runoob.com"))

# 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))

# 通过列表索引设置参数

my\_list = ['菜鸟教程', 'www.runoob.com']

print("网站名：{0[0]}, 地址 {0[1]}".format(my\_list)) # "0" 是必须的