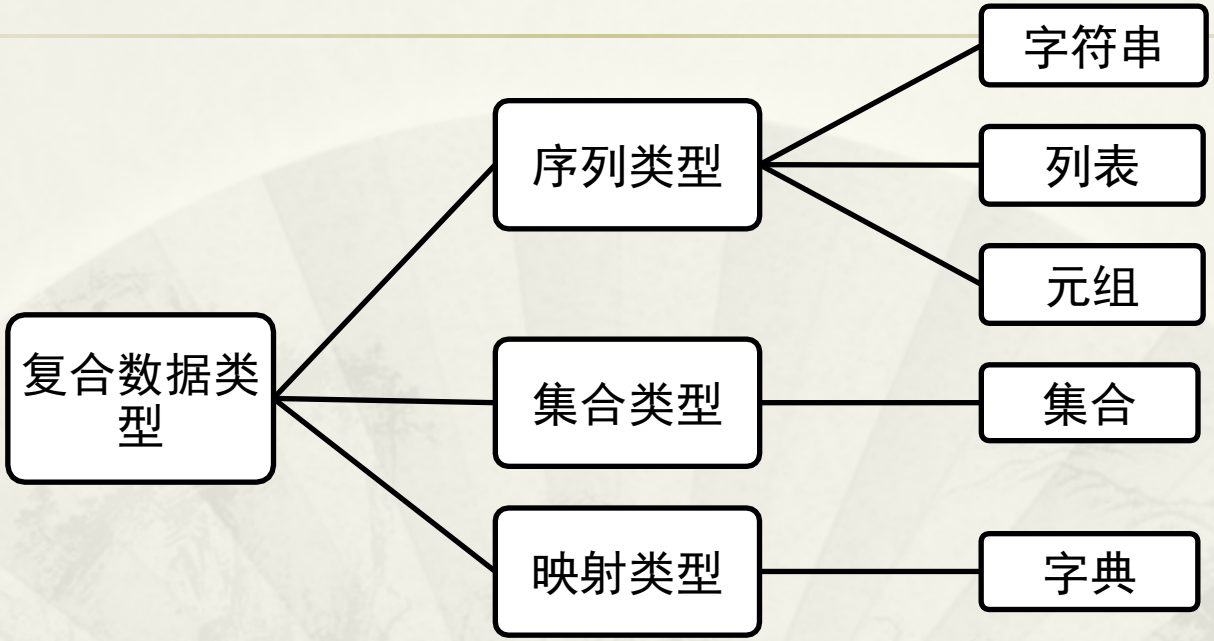
第6章 复合数据类型

第6章 复合数据类型

6.1 序列的通用操作
6.2 列表
6.3元组
6.4 集合
6.5 字典

复合数据类型概述



序列类型是一个元素向量，元素之间存在先后关系，通过序列中的索引 (位置编号)来访问数据元素，元素之间不排他可重复。

映射类型是“键-值”数据项的组合，每个元素是一个键值对，表示为(key,value)。

集合类型是一个元素集合，元素之间无序、不重复。

6.1 序列的通用操作

序列类型包括字符串(str)、列表(list)和元组(tuple)。

```
a= “Monkey“                          #字符串
b==['Alice','Ben','Candy','Mary','Lucy’]    #列表
c=(=(“Rat”，“OX”，“Tiger”，“Rabbit”，“Dragon”，“Sn
```

序列类型都可以进行索引、切片、序列相加、序列相乘以及成员资格检查等操作。

Python还有计算序列长度、找出最大元素和最小元素等内置函数。

6.1.1 序列的索引

- **序列名[索引]**
- 有两种索引方式：正向索引和反向索引

例如：

```
>>> str1="this is a string"
>>> Lst1=["2022100123","张三","男",18]
>>> tup1=("物理","化学","地理","生物","历史")
>>> print(str1[0],Lst1[-1],tup1[3])
t 18 生物
```

6.1.2 序列的切片

- 切片就是取出序列中某个范围内的元素，得到一个新的序列，原序列不变。一般格式为：
- **序列名[M:N:K]**

```
>>> Lst2=[1,2,3,4,5,6,7,8,9]
>>> Lst2[1:4]    #步长缺省为1，索引[1,4) 结果： [2, 3, 4]
>>>Lst2[:]       #同lst2[::],步长缺省为1，取出从前往后所有元素[1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>Lst2[::-1]   #步长为-1，取出从后往前所有元素[9, 8, 7, 6, 5, 4, 3, 2, 1]
>>> Lst2[:-2:2]  #步长为2,M缺省为0,取出从[0,-2）步长为2的元素[1, 3, 5, 7]
>>> Lst2[3::2]  #步长为2,取出从3到末尾的步长为2的元素 [4, 6, 8]
>>> lst2[-1::-2] #步长为-2,N为第一个元素，取出从-1到0步长为-2的元素[9, 7, 5, 3, 1]
```

6.1.3 序列的计算

1. 序列相加

使用加号+可以进行同类型序列的连接操作。例如：

```
>>> [1,2,3]+['a','b','c']     #列表相加
[1, 2, 3, 'a', 'b', 'c']
>>> (4,5,6)+(7,8,9)           #元组相加
(4, 5, 6, 7, 8, 9)
>>> "abc"+"123"               #字符串相加
'abc123'
```

6.1.3 序列的计算

2. 序列相乘

用整数n乘以序列会产生一个新的序列，在新序列中，原序列将重复n次。当n<1,将返回空序列。例如：

```
>>> a=(1,2,3)
>>> 2*a
(1, 2, 3, 1, 2, 3)
>>> -1*a
()
```

6.1.3 序列的计算

3. 成员资格检查

成员资格检查用于检查一个值是否在序列中，有in和not in两个运算符，返回True或False。例如：

```
>>> tup1=("物理","化学","地理","生物","历史")
>>> "音乐" in tup1
False
>>> "音乐" not in tup1
True
```

6.1.4 序列处理函数

1.len()、max()、min()函数

len(s):返回序列s的元素个数

min(s):返回序列s的最小元素

max(s):返回序列s的最大元素

以下Python表达式中，哪项的值与其它三项不同（ ）。

A． len("Nice to meet you".split())　　B．len("Nice".split())

C．sum（[1，2，1，0]）

D．max（[1，2，3，4]）

2. index方法和count方法

s.index(x[,i[,j]])：返回序列s中索引为第i到第j-1项元素中第一次出现元素x的位置。

s.count(x)：返回序列s中出现x的次数。

6.2 列表

- 列表（list）是Python中最常用的复合数据类型，可以完成大多数复合数据结构的操作。
- 列表元素是可变的，也就是列表中的元素的值可修改，也可以增加元素或删除元素。
- 字符串和元组的元素是不可变的。所以除了序列的通用操作外，列表还有许多专有的操作

6.2.1 列表的创建

1. 用[]创建列表

列表是用方括号“[]”括起来、用逗号分隔的元素序列

```
>>> elist=[]       #创建空列表
>>> names=['Alice','Ben','Candy','Mary','Lucy']
>>> stud=["2022100212","张三",18,[88,77,89]]
>>> print("{}的三门课的成绩是{},{},{}".format(stud[1],stud[3][0],stud[3][1],stud[3][2]))
张三的三门课的成绩是88,77,89
>>> names[::2]            #通过切片访问列表中的元素
['Alice', 'Candy', 'Lucy']
```

6.2.1 列表的创建

- 2.用list()函数创建列表
- 列表也可以用list() 函数将range()对象、元组、字符串或其他可迭代类型的数据转换为列表。

```
>>> numlist=list(range(1,6))   #创建一个包含1到5的整数列表
>>> numlist
[1, 2, 3, 4, 5]
>>> strlist=list("hello")      #将字符串转为列表
>>> strlst
['h', 'e', 'l', 'l', 'o']
>>> tlist=list((1,2,3,4))      #将元组(1,2,3,4)转为列表
>>> tlist
[1, 2, 3, 4]
>>> elist2=list()              #创建空列表
>>> elist2
[]
```

列表的注意事项：

- 列表必须通过显示的数据赋值才能生成，简单将一个列表变量赋值给另一个列表变量不会生成新的列表

```
>>> lst1=[70,80,90]
>>> lst2=lst1
>>> id(lst1),id(lst2)
(50784384, 50784384)
>>> lst2[0]=72
>>> lst1
[72, 80, 90]
```

- 通过切片操作可以生成新的列表，不会改变原来的列表

```
>>> x=[1,2,3,4,5,6]
>>> y=x[:]
>>> y
[1, 2, 3, 4, 5, 6]
>>> id(x),id(y)
(51019904, 51370112)
```

6.2.2 列表的专有操作

1.列表元素的添加

ls.append(x)方法：在列表末尾追加元素

ls.insert(i,x)方法：在列表第i个位置增加新元素x

ls.extend(lst)方法：将列表lst元素追加到列表ls尾部

“+=”运算符： ls+=lst将列表lst元素追加到列表ls尾部

“*=”运算符：ls*=n更新列表ls，其元素重复n次

append、insert、extend、+=、*=这些操作在添加了新元素后，内存地址没有发生变化，属于原地操作，不会创建新的列表对象

```
>>> names=['Alice','Ben']
>>> id(names)
51371264
>>> names.append("Yoyo")          #['Alice', 'Ben', 'Yoyo']
>>> names.insert(2,"Lily")    #['Alice', 'Ben', 'Lily', 'Yoyo']
>>> n=["张三","李四"]
>>> names.extend(n)
        #['Alice', 'Ben', 'Lily', 'Yoyo', '张三', '李四']
>>> n1=["王五"]
>>> names+=n1     #与names=names+n1不等价
   #['Alice', 'Ben', 'Lily', 'Yoyo', '张三', '李四', '王五']
>>> names*=2
#结果为：['Alice', 'Ben', 'Lily', 'Yoyo', '张三', '李四', '王五',
'Alice', 'Ben', 'Lily', 'Yoyo', '张三', '李四', '王五']
>>> id(names)
51371264
```

下列Python程序的运行结果是（　）。

```
s=[1,2,3,'a']
s.append([4,5])
print(len(s))
```

A．2　　B．4　　C．5　　D．6

6.2.2 列表的专有操作

2. 列表元素的删除

del语句

- del ls：删除列表ls对象
- del ls[i]：删除列表ls第i项的元素
- del ls[i:j:k]：删除列表ls第i到第j-1项以k为步长的元素

ls.pop(i)方法：返回列表ls第i项元素并删除该元素，缺省i删最后一个

ls.remove(x)方法：将列表中第一个出现的元素x删除

ls.clear()方法：　删除列表ls中的所有元素，把ls变为空列表

6.2.2 列表的专有操作

```
>>> names=['Alice','Ben','Candy','Mary','Lucy']
```

```
>>> names=['Alice','Ben','Candy','Mary','Lucy']
>>> names.clear()
>>> names
[]
```

以下代码的输出结果是（　）。

```
list1=[1,2,3,4]
list2=list1
list2.clear()
print(list1)
```

A．[1,2,3,4]　　B．1,2,3,4　　C．[]　　D．错误

6.2.2 列表的专有操作

3. 列表元素的修改

可以通过索引或切片直接修改列表元素的值。

- ls[i]=x
- ls[i:j:k]=lst

当使用切片赋值将一个列表赋给另一个列表时，Python不要求两个列表长度一样，但遵循“多增少减”的原则。可以通过赋给更多或更少元素实现列表元素的插入或删除

6.2.2 列表的专有操作

```
>>> a=[1,2,3,4,5]
>>> a[2::]=[7,8,9,10]
        #把下标从2到所有的元素改为[7,8,9,10]
>>> a
[1, 2, 7, 8, 9, 10]
>>> a[0:2]=[6]      #把下标为0和1的元素改[6]
>>> a
[6, 7, 8, 9, 10]
```

6.2.2 列表的专有操作

4. 对列表进行排序和逆置

- ls.sort(key=None,reverse=False)： reverse=True表示降序排列，reverse参数缺省，或reverse=False表示升序排列。key参数用于指定一个函数，此函数将在每个元素比较前被调用。
- ls.reverse()： 反转逆置列表ls中的元素。
- 内置函数sorted(ls[,reverse=False])：将列表ls默认升序排列后生成的新列表返回，原列表ls的值不变。若reverse=True，表示降序。

6.2.2 列表的专有操作

内置函数sorted(ls[,reverse=False])的举例如下：

```
>>> s=[23,21,45,67,12,9]
>>> t=sorted(s)
>>> s
[23, 21, 45, 67, 12, 9]
>>> t
[9, 12, 21, 23, 45, 67]
```

字符串 str="cat dog tiger"，以下程序的输出结果是：（ ）。

```
str= "cat dog tiger"
ls=str.split()
ls.reverse()
print(ls)
```

A．'tiger', 'dog', 'cat’

B．tiger dog cat

C．None

D．['tiger', 'dog', 'cat']

6.2.3 遍历列表

- 1. 用for循环语句遍历列表

格式：　　for 变量名　in 列表名：
　　语句块

【例 6-1】用for循环遍历列表，将列表中的每个元素星期几的英文全称改为缩写，输出时用空格隔开。

```
week_list=["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]
i=0
for item in week_list:
    week_list[i]=item[0:3]
    i=i+1
for item in week_list:
    print(item,end=" ")
```

6.2.3 遍历列表

2. 用while循环语句遍历列表

- 用while循环语句遍历列表时，需要先计算列表的长度，将获得的长度作为循环的条件,用索引访问列表元素。

【例 6-2】用列表的方法输出斐波那契数列：1,1,2,3,5,8…前10项，并计算这10项的和。

```
lst=[1,1]
i=2
while i<=10:
    lst.append(lst[i-1]+lst[i-2])
    i=i+1
print(lst)
i=0
s=0
while i<len(lst):
    s+=lst[i]
    i=i+1
print(s)
```

```
lst=[1,1]
i=2
while i<=10:
    lst.append(lst[i-1]+lst[i-2])
    i=i+1
print(lst)
print(sum(lst))
```

6.2.4 列表推导式

- 列表推导式可以利用range对象、元组、列表、字典和集合等数据类型，快速生成一个满足条件的列表。语法格式如下：
- **[表达式 for 迭代变量 in 迭代对象 [if 条件表达式]]**
- 其中 [if 条件表达式]可缺省。将每个符合if条件的迭代变量计算一次表达式的值，这些值就是新生成的列表中的每个元素。

1.不加if条件的列表推导式

创建一个1到10的整数列表，代码如下：

```
>>> a=[x+1 for x in range(10)]
>>> a
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

>>> import random
>>> b=[random.randint(0,100) for i in range(10)]
>>> b
[76, 17, 98, 10, 56, 22, 43, 2, 66, 86]
```

2. 加if条件的列表推导式

创建一个1到20间奇数的平方的列表，代码如下：

```
>>> b=[x*x for x in range(1,21) if x%2==1]
>>> b
[1, 9, 25, 49, 81, 121, 169, 225, 289, 361]
```