

5.29总结：元组、集合、字典、推导式

tuple（元组）只读的列表

元组（**tuple**）是 Python 中一种不可变的序列类型，用于存储**有序、可重复**的元素。它与列表（list）类似，但关键区别在于 **元组一旦创建，内容不可修改**（不可变性）。

```
tuple1 = (1, 2, 34, 5, 6) #元组定义
print(type(tuple1)) #类型为tuple
print(tuple1[3]) #访问元素
for i in tuple1: #遍历元组
    print(i)

#想对元组操作，需要将其转为列表
list1=list(tuple1)
```

set（集合）无序，去掉重复数据

集合（**set**）是 Python 中一种**无序、不重复**的容器类型，基于哈希表实现，常用于快速成员检测、去重和数学集合运算（如并集、交集）。

```
set1 = set() #创建空集合，必须使用set(),使用{}会创建一个空字典
set1 = {1, 2, 3, 4, 5, 5, 4, 3, 2, 1} #集合定义
print(type(set1)) #类型为set
print(set1) #只有{1, 2, 3, 4, 5}
set1.add(666) #添加元素
set1.remove(55) #删除元素
set1.pop() #随机删除一个元素
#不能使用下标访问set，所以修改操作一般为remove操作 + add操作
```

dict（字典）

字典是 键 (key): 值 (value) 的集合
在同一个字典中，键 (key) 必须是唯一的
创建空字典使用 {}

```
# 创建字典
dict1 = {'name': 'Alice', 'age': 30, 'city': 'New York'} # 直接创建
dict2 = dict([('name', 'Bob'), ('age', 25)]) # 使用 dict() 函数从元组列表创建
dict3 = dict(name='Charlie', age=35) # 使用关键字参数创建
```

```

empty_dict = {} # 创建空字典

# 访问元素
print(dict1['name']) # 通过键获取值，键不存在会报错
print(dict1.get('age')) # 通过 get() 方法获取值，键不存在返回 None
print(dict1.get('gender', 'Unknown')) # 可设置默认值，键不存在时返回

# 添加/修改元素
dict1['job'] = 'Engineer' # 若键不存在，会添加新键值对
dict1['age'] = 31 # 若键存在，则修改键对应的值
dict1.update({'city': 'Los Angeles', 'hobby': 'Reading'}) # 批量添加/更新

# 删除元素
del dict1['city'] # 删除指定键的键值对
age = dict1.pop('age') # 删除指定键的键值对，并返回对应的值
last_item = dict1.popitem() # 删除并返回字典的最后一个键值对（Python 3.7+ 有序）
dict1.clear() # 清空字典

# 遍历字典
for key in dict1: # 遍历键
    print(key, dict1[key])
for value in dict1.values(): # 遍历值
    print(value)
for key, value in dict1.items(): # 遍历键值对
    print(key, value)

# 其他常用操作
len(dict1) # 返回字典中键值对的数量
'name' in dict1 # 检查键是否存在
list(dict1.keys()) # 获取所有键的列表
list(dict1.values()) # 获取所有值的列表
list(dict1.items()) # 获取所有键值对的列表

# 字典复制
dict_copy = dict1.copy() # 浅复制
import copy
deep_copy = copy.deepcopy(dict1) # 深复制

# 字典合并（Python 3.9+）
merged_dict = dict1 | dict2 # 使用 | 运算符合并两个字典
dict1 |= dict2 # 原地合并（更新 dict1）

```

推导式

输入源： range list tuple set dict

输出源： list tuple set dict

推导式格式为：表达式 for 变量 in 输入源 if 条件

推导式格式为：表达式 for 变量 in 输入源 if 条件 for 变量 in 输入源 if 条件

```
# 给定一个列表,将每一位数字变成它的平方
# alist = [1, 2, 3, 4, 5, 6, 7]
# 输出结果: [1, 4, 9, 16, 25, 36, 49]
alist = [1, 2, 3, 4, 5, 6, 7]
print([i**2 for i in alist])
print([i**2 for i in alist if i+2>5])
print({i**2 for i in alist})
print(tuple((i**2 for i in alist)))
print({i**2:f"test{i}" for i in alist})
#推导 30 以内可以被 3 整除的整数为列表:
print([i for i in range(30) if i % 3 == 0])
```

数据源使用字典

```
d1={"张三":20,"李四":30,"王五":40}
# 推导列表
print([i for i in d1.values()])
# 推导字典
print({i:i**2 for i in d1.values() if i>20})
# 推导集合
print({i for i in d1.values() if i>20})
# 推导元组
print(tuple((i for i in d1.values() if i>20)))
# 二重循环 提取数字为列表[45,67,45,34,65,78]
l1=[[45,67,45,"hhh"],[34,65,78,"test"]]
print([j for i in l1 for j in i if type(j)==int])
```

作业练习

```
#1. 获取字典中的值: 获取下列字典键为 'history' 的值
sampleDict = {"class":{"student":{"name":"Mike","marks":
{"physics":70,"history":80}}}}
print(sampleDict["class"]["student"]["marks"]["history"])

# 2. 请将元组 v1 = (11,22,33) 中的所有元素追加到列表 v2 = [44,55,66] 中
v1 = (11,22,33)
v2 = [44,55,66]
# v2=v2+list(v1)
```

```

v2+=v1
print(v2)

# 3. 已知列表:
list1 = ['life', 'is', 'short']
list2 = ['you', 'need', 'python']
list3 = [1, 2, 3, 4, 5, 3, 4, 2, 1, 5, 7, 9]
#完成以下操作:
# (1) 把上述三个列表构造为一个集合 set1
set1 = set(list1+list2+list3)
print(set1)
# (2) 输出集合的长度
print(len(set1))
# (3) 把 python 从集合中移除
set1.remove('python')
print(set1)

# 4. 已知列表
li = [11, 22, 100, 33, 44, 66, 55, 77, 88, 99, 90]
# 将所有大于66的值保存至字典的第一个key对应的列表中, 将小于66的值保存至第二个key对应的列表
# 中。
d1 = {'key1': [], 'key2': []}
for i in li:
    if i > 66 :
        d1['key1'].append(i)
    elif i < 66 :
        d1['key2'].append(i)
print(d1)
# 输出结果: {'key1': [100, 77, 88, 99, 90], 'key2': [11, 22, 33, 44, 55]}

# 5. 已知列表
list1 = [11, 22, 11, 33, 44, 55, 66, 55, 66]
#统计列表中每个元素出现的次数, 生成一个字典 输出结果: {11:2, 22:1, ...} 注: 顺序不要求
d1 = {}
for i in list1:
    if i not in d1:
        d1[i]=1
    else:
        d1[i]+=1
print(d1)

# 6. 已知字典
d1={"subjects":[{"rate":"7.4", "cover_x":640}, {"rate":"6.0", "cover_x":1080}]}
# 处理显示成如下格式输出: 7.4 640 6.0 1080
l1=[]
for i in d1.values():

```

```

    for j in i:
        for k in j.values():
            l1.append(k)
print(*l1)

# 7. 已知如下数据
d1 = { '192.168.1.1':{'cpu':'0.23', '内存':'16', '硬盘':'500'},
        '192.168.1.2':{'cpu':'3.22', '内存':'64', '硬盘':'700'},
        '192.168.1.3':{'cpu':'1.99', '内存':'32', '硬盘':'800'} }

#处理显示成如下格式输出:
# 192.168.1.1: cpu 0.23 内存 16 硬盘 500
# 192.168.1.2: cpu 3.22 内存 64 硬盘 700
# 192.168.1.3: cpu 1.99 内存 32 硬盘 800
for i,j in d1.items():
    str1=f"{i}: "
    for n,m in j.items():
        str1+=f"{n} {m}"
    print(str1)

# 8. 有字符串 "k: 1|k1:2|k2:3 |k3 :4" 处理成字典 {'k':1,'k1':2,'k3':4}
# 注: 字符串替换使用 replace 函数
str1 = "k: 1|k1:2|k2:3 |k3 :4"
str1 = str1.replace(" ", "")
list1 = str1.split("|")
dict1 = {}
for i in list1:
    key,value = i.split(":")
    if key != "k2":
        dict1[key] = int(value)
print(dict1)

# 上机练习4
# 1. 生成一个存放 1-100 之间个位数为 3 的数据列表
print([i for i in range(1,101) if (i)%10==3])

# 2. 利用列表推导式将已知列表中的整数提取出来 [17, 98, 34, 21]
list_two = [True, 17, "hello", "bye", 98, 34, 21]
print([i for i in list_two if type(i)==int])

# 3. 根据列表利用推导式存放指定列表中 字符串的长度如下
# {'good':4, 'nice':4, 'see you':7, 'bye':3}
list_three=["good", "nice", "see you", "bye"]
print({i:len(i) for i in list_three})

# 4. 生成一个列表, 其中的元素为 '0-1', '1-2', '2-3', '3-4', '4-5'
print([f"{i}-{i+1}" for i in range(0,5)])

```

```
# 5. 已知数据
t1=((1,100),(2,30),(3,80),(4,234))
# 推导成列表 [100,30,80,234]
print([i[1] for i in t1])
# 推导成字典 {1: 100, 2: 30, 3: 80, 4: 234}
print({i[0]:i[1] for i in t1})
# 推导成列表 [{'name':1, 'value':100},{'name':2,'value':30},
{'name':3,'value':80},{'name':4,'value':234}]
print([{'name':i[0],'value':i[1]} for i in t1])
```

6. 已知如下列表 `students`，在列表中保存了 6 个学生的信息，根据要求完成下面的题目

```
students = [
    {'name': '小花', 'age':19, 'score':90, 'gender': '女', 'tel': '15300022839'},
    {'name': '明明', 'age':20, 'score':40, 'gender': '男', 'tel': '15300022838'},
    {'name': '华仔', 'age':18, 'score':100, 'gender': '女', 'tel': '15300022839'},
    {'name': '静静', 'age':16, 'score':90, 'gender': '不明', 'tel': '15300022428'},
    {'name': 'Tom', 'age':17, 'score':59, 'gender': '不明', 'tel': '15300022839'},
    {'name': 'Bob', 'age':18, 'score':90, 'gender': '男', 'tel': '15300022839'} ]
```

#打印学生分数列表

```
print([i['score'] for i in students])
```

#打印手机尾号是 8 的学生的名字列表

```
print([i['name'] for i in students if i['tel'][-1]=='8'])
```

#打印不及格的同学的所有信息列表，效果如下：

```
print([j for i in students if i['score']<60 for j in i.values()])
#['明明',20,40,'男','15300022838','Tom',17,59,'不明','15300022839']
```