

# python 语言程序设计基础

Hengsheng Zhou

电信与智能制造学院

2025 年 5 月 8 日



郑州西亚斯学院  
SIAS UNIVERSITY

# 回顾

请同志们先思考一下列表的特性

# 元组

## 元组的创建

列表	元组
有序	有序
有重复元素	有重复元素
元素值可更改	元素值可更改
可增删元素	不可增删元素

# 元组

## 元组的创建

列表	元组
有序	有序
有重复元素	有重复元素
元素值可更改	元素值可更改
可增删元素	不可增删元素

```
mytuple1=(1,2,3,4,5)  
# 使用中括号创建元组
```

```
1 mytuple2=tuple({1,2,3,4})  
2 mytuple2=tuple((1,2,3,4))  
3 mytuple2=tuple({1,2,3,4})  
4 # 使用构造器创建元组
```

# 元组

## 元组的创建

列表	元组
有序	有序
有重复元素	有重复元素
元素值可更改	元素值可更改
可增删元素	不可增删元素

```
mytuple1=(1,2,3,4,5)  
# 使用中括号创建元组
```

```
1 mytuple2=tuple({1,2,3,4})  
2 mytuple2=tuple((1,2,3,4))  
3 mytuple2=tuple({1,2,3,4})  
4 # 使用构造器创建元组
```

## 定义单个元素的元组

```
thistuple = ("apple",)  
# 将创建一个元组  
thistuple = ("apple")  
# 将创建一个字符
```

# 练习

## 例 (1)

使用构造器创建一个具有 10000 个随机元素的元组

# 元组

## 访问元组

- 索引 (正向, 反向, 截取)
- 遍历
- 筛选

# 元组

## 访问元组

- 索引 (正向, 反向, 截取)
- 遍历
- 筛选

### 例 (1)

访问元组的所有索引值为偶数的元素



# 元组

添加元素/删除元素

因为 tuple 是不可更改的，如果需要更改 tuple 中的元素需要将其转化为 list 类型的变量

# 元组

## 解包

将元组中的元素一次赋给多个变量

### example

```
fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")  
(green, yellow, *red) = fruits  
print(green) print(yellow) print(red)
```

# 元组

## 解包

将元组中的元素一次赋给多个变量

### example

```
fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")  
(green, yellow, *red) = fruits  
print(green) print(yellow) print(red)
```

### 例 (1)

定义一个函数使用解包的方式为函数赋值

# 元组

## 方法

- `count()` : 输出某个元素在 tuple 中出现的次数
- `index()` : 输出某个元素在元组中第一次出现位置的索引值

# 集合 set

## 创建集合

列表	元组	集合
有序	有序	无序
有重复元素	有重复元素	无重复值
元素值可更改	元素值可更改	元素值不可变
可增删元素	不可增删元素	可增删元素

# 集合 set

## 创建集合

列表	元组	集合
有序	有序	无序
有重复元素	有重复元素	无重复值
元素值可更改	元素值可更改	元素值不可变
可增删元素	不可增删元素	可增删元素

### 例

1,2,3

set(list or set)

# 讨论

请同学们讨论一下 1 和 True 或 0 和 False 可以在同一个集合中么？

# 集合

访问 set

- 索引
- 遍历
- 筛选
- 查存

## 查存

```
thisset = "apple", "banana", "cherry"  
print("banana" in thisset)
```



# 集合

## 向集合中添加元素

- `add()`
- `update()` 更新原集合、`union()` 返回新集合

# 集合

## 删除集合中的元素

- `remove()`
- `discard()`
- `pop()`
- `clear()`

# 集合

## 集合中的方法

- `intersection()` 取两个集合的交集
- `intersection_update()` 在原集合上更改
- `difference()` 等同于  $A-B$  将在另一个集合中出现过的元素筛掉形成新集合
- `difference_update()` 在元集合上更改
- `symmetric_difference()` 等同于  $(A-B)+(B-A)$
- `symmetric_difference_update()` 在原集合上更改

# 字典

## 创建字典

列表	元组	集合	字典
有序	有序	无序	有序
有重复元素	有重复元素	无重复值	无重复值
元素值可更改	元素值可更改	元素值不可变	元素值可变
可增删元素	不可增删元素	可增删元素	可增删元素

# 字典

## 创建字典

列表	元组	集合	字典
有序	有序	无序	有序
有重复元素	有重复元素	无重复值	无重复值
元素值可更改	元素值可更改	元素值不可变	元素值可变
可增删元素	不可增删元素	可增删元素	可增删元素

字典是有序、可改值、不允许重复的集合

`{1,2,3,4}`

`dict(list or tuple or set)`

`.fromkeys('abc')`

# 字典

## 创建字典

列表	元组	集合	字典
有序	有序	无序	有序
有重复元素	有重复元素	无重复值	无重复值
元素值可更改	元素值可更改	元素值不可变	元素值可变
可增删元素	不可增删元素	可增删元素	可增删元素

字典是有序、可改值、不允许重复的集合

`{1,2,3,4}`

`dict(list or tuple or set)`

`.fromkeys('abc')`

### 例 (1)

生成一个具有 1000 个元素的字典，其中元素的值分别为 0-100 的随机值

# 字典

## 字典元素的访问

- `get()`: 通过 key 值访问某个元素的 value
- `keys()`: 返回所有的 keys
- `values()`: 返回所有的 values
- `items()`: 返回所有的 items
- 遍历字典
  - `for x in dictionary` 和 `keys()`: 遍历 keys
  - `for x in dictionary: dictionary[x]` 和 `values()` 便利 values
  - `for x,y in dictionary.items:` 遍历 (key,value)

# 讨论

请同学们讨论一下若使用 `keys` 获取到所有键位后更改了字典那么以获取的键位列表会不会改变?



# 字典

## 向字典中添加元素

- `dictionary[keys]=values`
- `update()`: 函数的值可以是任何 iterable 类型的变量

# 字典

## 删除字典中的元素

- `pop(key)` 删除指定 `key` 的元素
- `popitem()` 删除最后一个元素
- `clear()`