

- 容器
 - 列表
 - 元组
 - 字典
 - 集合

元组

创建元组

元组是有序的、不能更改的、可重复的容器。

example

创建元组 `thistuple = ("apple", "banana", "cherry", "apple", "cherry")` 使用构造器创建元组 `thistuple = tuple(("apple", "banana", "cherry", "apple", "cherry"))`

定义单个元素的元组

创建单元素的元组必须在元素后添加逗号 `thistuple = ("apple",)`
`print(type(thistuple))` `thistuple = ("apple")` `print(type(thistuple))`

元组

访问元组

- 索引 (正向, 反向, 截取)
- 便利
- 筛选

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

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

set 创建

set 是无序、不可更改、不可重复、无索引的容器

example

```
thisset = {"apple", "banana", "cherry"} print(thisset) 通过构造器创建  
set thisset = set(("apple", "banana", "cherry")) note the double  
round-brackets
```

set 元素不允许重复

```
thisset = {"apple", "banana", "cherry", True, 1, 2}  
1 和 true 在 set 中被认为是值相同的元素 print(thisset)
```

集合

访问 set

- 索引
- 遍历
- 筛选
- 检查 set 没有索引，索引、筛选操作需将 set 转化为 list

检查

```
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()` \$ 将两集合中所有重复的元素返回到新集合
- `intersection update()` 不返回新集合, 在原集合上更改
- `difference()` -将在另一个集合中出现过的元素筛掉形成新集合
- `difference_update()` 不创建新集合
- `symmetric_difference()`
 - ^ 两个集合中所有差异的元素全部同步到新集合
- `symmetric_difference_update()` 不创建新集合在原集合上更改

mind

使用运算符操作集合时, 只能操作 `set` 类型的值

字典

创建字典

字典是有序、可改值、不允许重复的集合

example

```
thisdict = {"brand": "Ford", "model": "Mustang", "year": 1964}
```

```
thisdict = dict(name = "John", age = 36, country = "Norway") : 使用构造器创建集合
```

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)

Attention

`x = dictionary.keys()`: 在获取字典的 keys 之后任何对字典的修改都会同步到 keys 列表, value 和 items 也类似

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()`

字典

字典的相关方法

- `fromkeys()`: 返回一个所有值相同的字典