

6. 容器

字典 / Dictionary

初始化

```
# 键值对用冒号分割, 包括在花括号中
> dict = {'a': 1, 'c': 2, 'b': '3'}
```

取值

```
> dict.items() # 取全部键值对
dict_items([('a', 1), ('c', 2), ('b', '3')])
> dict.keys() # 取全部的 key
dict_keys(['a', 'c', 'b'])
> dict.values() # 取全部的 values 值
dict_values([1, 2, '3'])
> dict['b'] # 取 key 为 b 对应的值
'3'
> dict['d'] # 取 key 为 d 对应的值, 将报错
KeyError: 'd'
```

赋值与修改

```
> dict['d'] = "ShowMeAI" # 赋值, 添加
> dict['c'] = 8 # 更新
```

删除

```
> del dict['c'] # 删除键是 'c' 的条目
```

元组 / Tuple

初始化

```
> tup1 = () # 空元组
> tup1 = (50,) # 只有一个元素的元组
> tup3 = ('physics', 'chemistry', 1997, 2000)
> tup4 = (1, 2, 3, 4, 5, 6, 7)
```

取值

```
> tup3[0] # 取 index 为 0 的值
> tup4[1:5] # 取 index 从 1 到 4 的值
```

修改

```
> tup1[0] = 100 # 错误! 元组的内容不允许更改
> tup_new = (100,) + tup1[1:] # 重新定义一个元组
```

删除

```
> del tup # 只能删除整个元组, 不可以删除某个元素
```

集合 / Set

初始化

```
# 集合 set 是无序不重复元素序列, 通过序列和 {} 定义
> company = {"Google", "Twitter", "Taobao"}
```

添加元素

```
> company.add("Facebook") # 添加

# 追加集合
> company.update({"Tencent", "Bytedance", "ShowMeAI"})
```

删除

```
> company.remove("ShowMeAI") # 如果没有这个值, 会报错
> company.discard("ShowMeAI") # 有这个值就删除, 没有就不操作
```

集合运算

```
> a = set('abracadabra')
> b = set('alacazam')
> a
{'a', 'r', 'b', 'c', 'd'}
> a - b # 集合 a 中包含而集合 b 中不包含的元素
{'r', 'd', 'b'}
> a | b # 集合 a 或 b 中包含的所有元素
{'a', 'c', 'r', 'd', 'b', 'm', 'z', 'l'}
> a & b # 集合 a 和 b 中都包含的元素
{'a', 'c'}
> a ^ b # 不同时包含于 a 和 b 的元素
{'r', 'd', 'b', 'm', 'z', 'l'}
```

计数器 / Counter

Counter 用于对容器进行计数

```
> from collections import Counter
# 颜色列表
> colors = ['blue', 'blue', 'blue', 'red', 'red']
> counter = Counter(colors) # Counter 计数
> counter['yellow'] += 1 # 频次 +1
Counter({'blue': 3, 'red': 2, 'yellow': 1})
> counter.most_common()[0] # 最高频
('blue', 3)
```

7. 流程控制

选择结构 - if 语句

```
> x = int(input("请输入一个整数: "))
> if x > 0:
    print(x, '是一个正整数')
> elif x < 0:
    print(x, '是一个负整数')
> else:
    print('0')
```

循环结构 - for/while

```
> words = ['show', 'me', 'ai', 'is', 'awesome']
> for w in words: # for 循环
    print(w, end=" ")
> show me ai is awesome

> count = 0
> while (count < 9): # while 循环
>     print('The count is:', count)
>     count = count + 1
```

8. 字典 / 列表推导式

```
# 对 0-9 的列表中每个值进行平方变换
> [x**2 for x in range(10)]
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
# 对 0-29 的列表中满足 3 的倍数的值进行平方变换
> [i**2 for i in range(30) if i % 3 is 0]
[0, 9, 36, 81, 144, 225, 324, 441, 576, 729]
```

```
> listdemo = ['ShowMeAI', 'show-me-ai.com']
```

```
# 对 listdemo 里每个值生成一个值: 长度的键值对, 构建成字典
> {key: len(key) for key in listdemo}
{'ShowMeAI': 8, 'show-me-ai.com': 14}
```

8. 遍历

range[示例]

```
> range(10)
range(0, 10)

> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

S h o w M e A I

Enumerate[示例]

```
> for i, el in enumerate("ShowMeAI"):
>     print(i, el)
0 S
1 h
2 o
3 w
4 M
5 e
6 A
7 I
```

9. 函数定义 / Function

模板

```
> def function_name( parameters ):
>     " 函数_文档字符串 "
>     function_suite
>     return [expression]
```

示例

```
> def my_abs(x):
>     if x >= 0:
>         return x
>     else:
>         return -x
```

10. 类 / Class [示例]

```
> class ShowMeAI:
>     ' 所有内容的基类 '
>     articleCount = 0

>     def __init__(self, title, length):
>         self.title = title
>         self.length = length
>         ShowMeAI.articleCount += 1

>     def displayCount(self):
>         print("Total Article", ShowMeAI.articleCount)

>     def displayArticle(self):
>         print("Title : " + str(self.title), "Length: " + str(self.length))
```

11. 正则表达式 /Regex

```
> import re # 正则模块使用
```

re.match 函数

re.match 尝试从字符串的起始位置匹配一个模式

```
> import re
> print(re.match('www', 'www.show-me-ai.com').span()) # 在起始位置匹配
> print(re.match('com', 'www.show-me-ai.com'))      # 不在起始位置匹配
输出 (0, 3) 和 None
```

re.search 方法

re.search 扫描整个字符串并返回第一个成功的匹配。

```
> import re
> line = "ShowMeAI is awesome website";
> searchObj = re.search( r'(.*) is (.*) .*', line, re.M|re.I)
> if searchObj:
>     print("searchObj.group() : ", searchObj.group())
>     print("searchObj.group(1) : ", searchObj.group(1))
>     print("searchObj.group(2) : ", searchObj.group(2))
> else:
>     print("Nothing found!!")
searchObj.group() : ShowMeAI is awesome website
searchObj.group(1) : ShowMeAI
searchObj.group(2) : awesome
```

检索和替换

Python 的 re 模块提供了 re.sub 用于替换字符串中的匹配项。

```
> import re
> phone = "2021-969-559 # 这是一个国外电话号码 "
```

删除字符串中的 Python 注释

```
> num = re.sub(r'#$.*$', "", phone)
> print(" 电话号码是 : ", num)
```

删除非数字 (-) 的字符串

```
num = re.sub(r'\D', "", phone)
print(" 电话号码是 : ", num)
```

电话号码是 : 2021-969-559

电话号码是 : 2021969559



扫码回复 “速查表”

下载最新全套资料