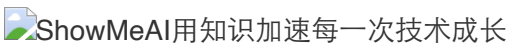


图解 Python 编程(17) | 集合

🕒 2021-11-09 👁 1175 💬 0 📁 工具教程 python 编程语言



作者：韩信子@ShowMeAI
教程地址：<https://www.showmeai.tech/tutorials/56>
本文地址：<https://www.showmeai.tech/article-detail/80>
声明：版权所有，转载请联系平台与作者并注明出处
收藏[ShowMeAI](#)查看更多精彩内容

Python集合

集合（Set）是一个无序的不重复元素序列。

可以使用大括号 `{ }` 或者 `set()` 函数创建集合，注意：创建一个空集合必须用 `set()` 而不是 `{ }`，因为 `{ }` 是用来创建一个空字典。

集合（Set）的4个特性

```
S = { 'ShowMeAI' , 2022 , 3.1415 }
```

- **无 序**：集合内的数据没有顺序
- **不 可 更 改**：集合内的元素不可更改
- **无类型要求**：集合内的元素可以是各种类型
- **唯 一 性**：集合不允许有重复的元素

<http://www.showmeai.tech/>

搜索 | 微信 ShowMeAI 研究中心

创建格式：

1. `parame = {value01,value02,...}`
2. 或者
3. `set(value)`

如下为示例代码（代码可以在在线python3环境中运行）：

```
1. >>> company = {'Baidu', 'ShowMeAI', 'google', 'ByteDance', 'ShowMeAI', 'Taobao', 'Tencent'}
2. >>> print(company)           # 这里演示的是去重功能
3. {'Baidu', 'ShowMeAI', 'google', 'ByteDance', 'Taobao', 'Tencent'}
4. >>> 'Baidu' in basket       # 快速判断元素是否在集合内
5. True
6. >>> 'Meituan' in basket
7. False
```

```
1. >>> # 下面展示两个集合间的运算.
2. ...
3. >>> a = set('abracadabra')
4. >>> b = set('alacazam')
5. >>> a
6. {'a', 'r', 'b', 'c', 'd'}
7. >>> a - b                   # 集合a中包含而集合b中不包含的元素
8. {'r', 'd', 'b'}
9. >>> a | b                   # 集合a或b中包含的所有元素
10. {'a', 'c', 'r', 'd', 'b', 'm', 'z', 'l'}
11. >>> a & b                   # 集合a和b中都包含了的元素
12. {'a', 'c'}
13. >>> a ^ b                   # 不同时包含于a和b的元素
14. {'r', 'd', 'b', 'm', 'z', 'l'}
```

类似列表推导式，同样集合支持集合推导式(Set comprehension)：

```
1. >>> a = {x for x in 'abracadabra' if x not in 'abc'}
2. >>> a
3. {'r', 'd'}
```

集合的基本操作

一个集合的实例

```
>>> name_1 = [1,2,3,4,7,8,7,10]

>>> name_1 = set(name_1)  #把列表转换为集合，转换后，去重


>>> print (name_1,type(name_1))

{1,2,3,4,7,8,10}, <class 'set'>
```

去重

转换为集合


搜索 | 微信 ShowMeAI 研究中心




ShowMeAI

集合 (Set) 的基本操作

关系测试	<code>name_1.intersection(name_2)</code>	<code>name_1.union(name_2)</code>	<code>name_1.difference(name_2)</code>	
	交集(<code>intersection()</code>)	并集(<code>union()</code>)	差集(<code>difference()</code>)	
	<code>name_3.issubset(name_1)</code>	<code>name_1.issuperset(name_3)</code>	<code>name_1.symmetric_difference(name_2)</code>	
	<code>name_1.isdisjoint(name_3)</code>			
	是否是子集	是否是父集	对称差集	
			是否有交集	
	关系测试 (特殊符号)	<code>name_1 & name_2</code>	<code>name_1 name_2</code>	<code>name_1 - name_2</code>
		交集(&)	并集()	差集(-)
		<code>name_1 ^ name_2</code>	<code>name_3 <= name_1</code>	<code>name_1 >= name_3</code>
		对称差集(^)	是否是子集(<=)	是否是父集(>=)
基本操作	<code>name_2.add(11)</code>	<code>name_2.update([12,13,14])</code>	<code>name_2.remove(1)</code>	
	添加(<code>add()</code>)	添加多项(<code>update()</code>)	删除	
	<code>name_2.discard(10)</code>	<code>len(name_1)</code>	<code>1 in name_1</code>	
	删除不存在的元素不会出现报错	长度(<code>len()</code>)	测试x是否是s的成员	

 <http://www.showmeai.tech/>

 搜索 | 微信

ShowMeAI 研究中心

(1) 添加元素

语法格式如下：

```
1. s.add( x )
```

将元素 x 添加到集合 s 中，如果元素已存在，则不进行任何操作。

```
1. >>> company = set(("Google", "ShowMeAI", "Taobao"))
2. >>> company.add("Facebook")
3. >>> print(company)
4. {'Taobao', 'Facebook', 'Google', 'ShowMeAI'}
```

还有一个方法，也可以添加元素，且参数可以是列表，元组，字典等，语法格式如下：

```
1. s.update( x )
```

x 可以有多个，用逗号分开。如下为示例代码（代码可以在在线python3环境中运行）：

```
1. >>> company = set(("Google", "ShowMeAI", "Taobao"))
2. >>> company.update({'Facebook', "LinkedIn"})
3. >>> print(company)
4. {'LinkedIn', 'Google', 'ShowMeAI', 'Facebook', 'Taobao'}
5. >>> company.update([1,4],[5,6])
6. >>> print(company)
7. {1, 3, 4, 5, 6, 'Google', 'Taobao', 'Runoob'}
```

(2) 移除元素

语法格式如下：

```
1. s.remove( x )
```

将元素 x 从集合 s 中移除，如果元素不存在，则会发生错误。如下为示例代码（代码可以在在线python3环境中运行）：

```
1. >>> company = set(("Google", "ShowMeAI", "Taobao"))
2. >>> company.remove("Taobao")
3. >>> print(company)
4. {'Google', 'ShowMeAI'}
5. >>> company.remove("Facebook") # 不存在会发生错误
6. Traceback (most recent call last):
7.   File "<stdin>", line 1, in <module>
8. KeyError: 'Facebook'
```

此外还有一个方法也是移除集合中的元素，且如果元素不存在，不会发生错误。格式如下所示：

```
1. s.discard( x )
```

```
1. >>> company = set(("Google", "ShowMeAI", "Taobao"))
2. >>> company.discard("Facebook") # 不存在不会发生错误
3. >>> print(company)
4. {'Taobao', 'Google', 'ShowMeAI'}
```

我们也可以设置随机删除集合中的一个元素，语法格式如下：

```
1. s.pop()
```

```
1. company = set(("Google", "ShowMeAI", "Taobao", "Facebook"))
2. x = company.pop()
3.
4. print(x)
```

输出结果：

```
1. ShowMeAI
```

多次执行测试结果都不一样。

set 集合的 pop 方法会对集合进行无序的排列，然后将这个无序排列集合的左面第一个元素进行删除。

(3) 计算集合元素个数

语法格式如下：

```
1. len(s)
```

计算集合 s 元素个数。

```
1. company = set(("Google", "ShowMeAI", "Taobao", "Facebook"))
2. print(len(company))
```

(4) 清空集合

语法格式如下：

```
1. s.clear()
```

清空集合 s。

```
1. company = set(("Google", "ShowMeAI", "Taobao", "Facebook"))
2. company.clear()
```

(5) 判断元素是否在集合中存在

语法格式如下：

```
1. x in s
```

判断元素 x 是否在集合 s 中，存在返回 True，不存在返回 False。

```
1. company = set(("Google", "ShowMeAI", "Taobao", "Facebook"))
2. "Facebook" in company
```

(6) 集合内置方法完整列表

方法	描述
add()	为集合添加元素
clear()	移除集合中的所有元素
copy()	拷贝一个集合
difference()	返回多个集合的差集
difference_update()	移除集合中的元素， 该元素在指定的集合也存在。
discard()	删除集合中指定的元素
intersection()	返回集合的交集
intersection_update()	返回集合的交集。
isdisjoint()	判断两个集合是否包含相同的元素， 如果没有返回 True， 否则返回 False。
issubset()	判断指定集合是否为该方法参数集合的子集。
issuperset()	判断该方法的参数集合是否为指定集合的子集
pop()	随机移除元素
remove()	移除指定元素
symmetric_difference()	返回两个集合中不重复的元素集合。
symmetric_difference_update()	移除当前集合中在另外一个指定集合相同的元素， 并将另外一个指定集合中不同的元素插入到当前集合中。
union()	返回两个集合的并集
update()	给集合添加元素

视频教程

也可以点击 [这里](#) 到B站查看有【中英字幕】的版本

【双语字幕+资料下载】Python 3全系列...

去bilibili观看

分享

扫一扫 手机看



00:00 / 14:19

360P

进入bilibili,一起发弹幕吐槽!

去吐槽


一键运行所有代码

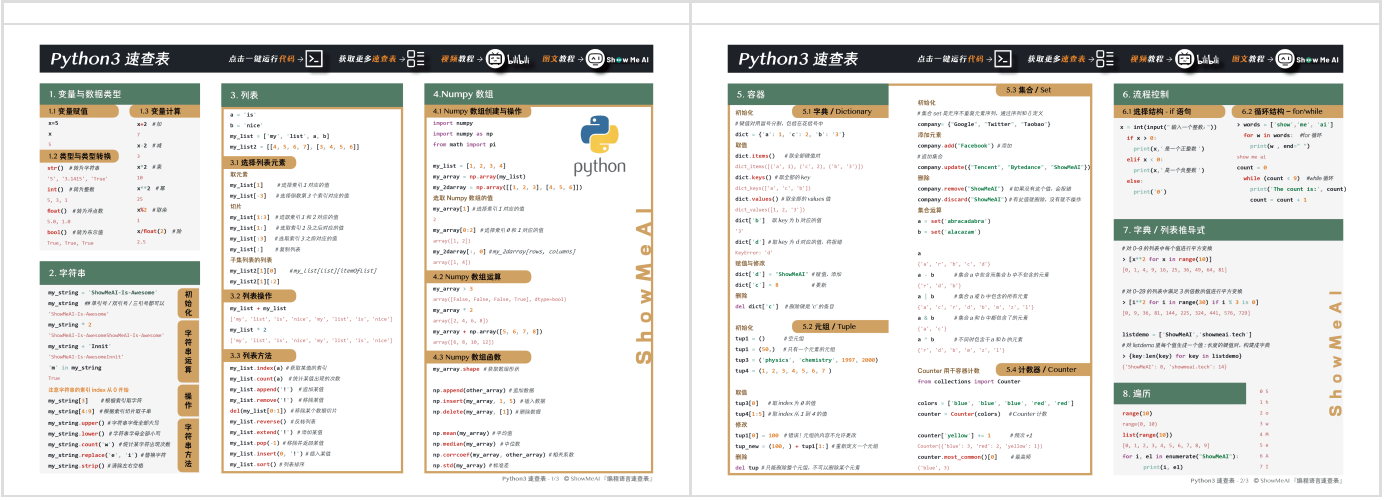
图解Python编程系列 配套的所有代码，可前往ShowMeAI 官方 **GitHub**，下载后即可在本地 Python 环境中运行。能访问 Google 的宝宝也可以直接借助 Google Colab一键运行与交互学习！

下载Python要点速查表

Awesome cheatsheets | **ShowMeAI**速查表大全 系列包含『编程语言』『AI技能知识』『数据科学工具库』『AI垂直领域工具库』四个板块，追平到工具库当前最新版本，并跑通了所有代码。点击 [官网](#) 或 [GitHub](#) 获取~

 ShowMeAI速查表大全

 Python 速查表（部分）



拓展参考资料

- Python教程 - Python3文档
- Python教程 - 廖雪峰的官方网站

ShowMeAI图解Python编程系列推荐（要点速查版）

- ShowMeAI 图解 Python 编程(1) | 介绍
- ShowMeAI 图解 Python 编程(2) | 安装与环境配置
- ShowMeAI 图解 Python 编程(3) | 基础语法
- ShowMeAI 图解 Python 编程(4) | 基础数据类型
- ShowMeAI 图解 Python 编程(5) | 运算符
- ShowMeAI 图解 Python 编程(6) | 条件控制与if语句
- ShowMeAI 图解 Python 编程(7) | 循环语句
- ShowMeAI 图解 Python 编程(8) | while循环
- ShowMeAI 图解 Python 编程(9) | for循环
- ShowMeAI 图解 Python 编程(10) | break语句
- ShowMeAI 图解 Python 编程(11) | continue语句
- ShowMeAI 图解 Python 编程(12) | pass语句
- ShowMeAI 图解 Python 编程(13) | 字符串及操作
- ShowMeAI 图解 Python 编程(14) | 列表
- ShowMeAI 图解 Python 编程(15) | 元组
- ShowMeAI 图解 Python 编程(16) | 字典
- ShowMeAI 图解 Python 编程(17) | 集合
- ShowMeAI 图解 Python 编程(18) | 函数
- ShowMeAI 图解 Python 编程(19) | 迭代器与生成器
- ShowMeAI 图解 Python 编程(20) | 数据结构
- ShowMeAI 图解 Python 编程(21) | 模块
- ShowMeAI 图解 Python 编程(22) | 文件读写
- ShowMeAI 图解 Python 编程(23) | 文件与目录操作
- ShowMeAI 图解 Python 编程(24) | 错误与异常处理
- ShowMeAI 图解 Python 编程(25) | 面向对象编程
- ShowMeAI 图解 Python 编程(26) | 命名空间与作用域
- ShowMeAI 图解 Python 编程(27) | 时间和日期

ShowMeAI系列教程精选推荐

- 大厂技术实现：推荐与广告计算解决方案
- 大厂技术实现：计算机视觉解决方案
- 大厂技术实现：自然语言处理行业解决方案
- 图解Python编程：从入门到精通系列教程
- 图解数据分析：从入门到精通系列教程
- 图解AI数学基础：从入门到精通系列教程
- 图解大数据技术：从入门到精通系列教程
- 图解机器学习算法：从入门到精通系列教程
- 机器学习实战：手把手教你玩转机器学习系列
- 深度学习教程：吴恩达专项课程·全套笔记解读
- 自然语言处理教程：斯坦福CS224n课程·课程带学与全套笔记解读
- 深度学习与计算机视觉教程：斯坦福CS231n·全套笔记解读

图解 Python 编程(17) | 集合

◀ 上一篇

图解 Python 编程(16) | 字典

下一篇 ▶

图解 Python 编程(18) | 函数