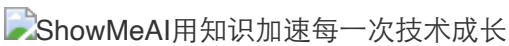


图解 Python 编程(14) | 列表

🕒 2021-11-09 👁 1647 💬 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/77>
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列表(List)

序列是Python中最基本和常见的数据结构。序列中的每个元素都分配一个数字 - 【它的位置，或索引】，第一个索引是0，第二个索引是1，依此类推。

序列都可以进行的操作包括索引，切片，加，乘，检查成员。

此外，Python已经内置确定序列的长度以及确定最大和最小的元素的方法。

列表是最常用的Python数据类型，它可以作为一个方括号内的逗号分隔值出现。

列表的数据项不需要具有相同的类型。

创建一个列表，只要把逗号分隔的不同的数据项使用方括号括起来即可。如下所示：

1. list1 = ['python', 'ShowMeAI', 1997, 2022]

2. list2 = [1, 2, 3, 4, 5]

3. list3 = ["a", "b", "c", "d"]

与字符串的索引一样，列表索引从0开始。列表可以进行截取、组合等。

AI
Show Me AI

列表 (List)

list3 = ['S', 'h', 'o', 'w', 'M', 'e', 'A', 'I']

索引

-8-7-6-5-4-3-2-1

01234567

list3[1:3]

['S', 'h']

list3[:4]

['S', 'h', 'o', 'w']

搜索 | 微信

ShowMeAI 研究中心

http://www.showmeai.tech/

AI
Show Me AI

列表 (List)

list3 = ['S', 'h', 'o', 'w', 'M', 'e', 'A', 'I']

截取

:

1234567:

list3[6:]

['A', 'I']

list3[:]

['S', 'h', 'o', 'w', 'M', 'e', 'A', 'I']

搜索 | 微信

ShowMeAI 研究中心

http://www.showmeai.tech/

访问列表中的值

使用下标索引来访问列表中的值，同样你也可以使用方括号的形式截取子列表。

AI
Show Me AI

使用下索引访问列表中的值

list1 = ['python', 'ShowMeAI', '1997', '2002']

反向索引

-4-3-2-1

正向索引

0123

list1[0]

搜索 | 微信

ShowMeAI 研究中心

http://www.showmeai.tech/

AI
Show Me AI

使用下索引访问列表中的值

图解Python 编程(14) | 列表

使用下索引访问列表中的值

list2 = [1 , 2 , 3 , 4 , 5 , 6 , 7]

0123456

list2[1:5]

List[1:5] = [2 , 3 , 4 , 5]

0123

搜索 | 微信 ShowMeAI 研究中心

http://www.showmeai.tech/

如下为示例代码（代码可以在在线python3环境中运行）：

```
1. list1 = ['python', 'ShowMeAI', 1997, 2022]
2. list2 = [1, 2, 3, 4, 5, 6, 7]
3.
4. print("list1[0]: ", list1[0])
5. print("list2[1:5]: ", list2[1:5])
```

以上代码执行结果：

```
1. list1[0]: python
2. list2[1:5]: [2, 3, 4, 5]
```

使用下索引访问列表中的值

list = ['red' , 'green' , 'blue' , 'yellow' , 'white', 'black']

反向索引-6-5-4-3-2-1

正向索引012345

list[-1]black

list[:-2]redgreenblueyellowwhite

list[-3:]yellowwhiteblack

搜索 | 微信 ShowMeAI 研究中心

http://www.showmeai.tech/

如下为示例代码（代码可以在在线python3环境中运行）：

```
1. list = ['red', 'green', 'blue', 'yellow', 'white', 'black']
2. print( list[-1] )
3. print( list[:-2] )
4. print( list[-3:] )
```

运行结果

```
1. black
2. ['red', 'green', 'blue', 'yellow']
3. ['yellow', 'white', 'black']
```

更新列表

你可以对列表的数据项进行修改或更新，你也可以使用append()方法来添加列表项，如下所示（代码可以在在线python3环境中运行）：

```
1. list = []      # 空列表
2. list.append('Google') # 使用 append() 添加元素
3. list.append('ShowMeAI')
4. print(list)
```

以上代码执行结果：

```
1. ['Google', 'ShowMeAI']
```

删除列表元素

可以使用 del 语句来删除列表的元素，如下所示（代码可以在在线python3环境中运行）：

```
1. list1 = ['python', 'ShowMeAI', 1997, 2022]
2. print(list1)
3. del list1[2]
4.
5. print("删除索引为2的元素后：")
6. print(list1)
```

以上代码执行结果：

```
1. ['python', 'ShowMeAI', 1997, 2022]
2. 删除索引为2的元素后：
3. ['python', 'ShowMeAI', 2022]
```

注意：我们后续会讨论remove()方法的使用

Python列表脚本操作符

列表对 `+` 和 `*` 的操作符与字符串相似。`+` 号用于组合列表，`*` 号用于重复列表。

如下所示：

Python 表达式	结果	描述
<code>len([1, 2, 3])</code>	<code>3</code>	长度
<code>[1, 2, 3] + [4, 5, 6]</code>	<code>[1, 2, 3, 4, 5, 6]</code>	组合
<code>['Hi!'] * 4</code>	<code>['Hi!', 'Hi!', 'Hi!', 'Hi!']</code>	重复
<code>3 in [1, 2, 3]</code>	<code>True</code>	元素是否存在于列表中
<code>for x in [1, 2, 3]: print x,</code>	<code>1 2 3</code>	迭代

Python列表截取

Python 的列表截取实例如下：

```
1. >>>L = ['Google', 'ShowMeAI', 'Baidu']
2. >>> L[2]
3. 'Baidu'
4. >>> L[-2]
5. 'ShowMeAI'
6. >>> L[1:]
7. ['ShowMeAI', 'Baidu']
8. >>>
```

描述：

Python 表达式	结果	描述
<code>L[2]</code>	<code>'Baidu'</code>	读取列表中第三个元素
<code>L[-2]</code>	<code>'ShowMeAI'</code>	读取列表中倒数第二个元素
<code>L[1:]</code>	<code>['ShowMeAI', 'Baidu']</code>	从第二个元素开始截取列表

Python列表函数方法

Python包含以下函数:

序号	函数	作用
1	<code>len(list)</code>	列表元素个数
2	<code>max(list)</code>	返回列表元素最大值
3	<code>min(list)</code>	返回列表元素最小值
4	<code>list(seq)</code>	将元组转换为列表

```
1. # 示例1：长度
2. list1, list2 = [123, 'ShowMeAI', 'google'], [456, 'abc']
3.
4. print("第1个列表长度：", len(list1))
5. print("第2个列表长度：", len(list2))
```

结果

```
1. 第1个列表长度： 3
2. 第2个列表长度： 2
```

```
1. # 示例2：最大最小值
2. list1, list2 = ['Baidu', 'ShowMeAI', 'google'], [456, 789, 200]
3.
4. print("第1个列表最大值：", max(list1))
5. print("第1个列表最小值：", min(list1))
6. print("第2个列表最大值：", max(list2))
7. print("第2个列表最小值：", min(list2))
```

结果

```
1. 第1个列表最大值： google
2. 第1个列表最小值： Baidu
3. 第2个列表最大值： 789
4. 第2个列表最小值： 200
```

```
1. # 示例3：转列表
2. aTuple = (123, 'ShowMeAI', 'google', 'Baidu');
3. aList = list(aTuple)
4.
5. print("列表元素：")
6. print(aList)
```

结果

```
1. 列表元素：
2. [123, 'ShowMeAI', 'google', 'Baidu']
```

Python包含以下方法:

序号	方法	作用
1	list.append(obj)	在列表末尾添加新的对象
2	list.count(obj)	统计某个元素在列表中出现的次数
3	list.extend(seq)	在列表末尾一次性追加另一个序列中的多个值 (用新列表扩展原来的列表)
4	list.index(obj)	从列表中找出某个值第一个匹配项的索引位置
5	list.insert(index, obj)	将对象插入列表
6	list.pop([index=-1])	移除列表中的一个元素（默认最后一个元素）， 并且返回该元素的值
7	list.remove(obj)	移除列表中某个值的第一个匹配项
8	list.reverse()	反向列表中元素
9	list.sort(cmp=None, key=None, reverse=False)	对原列表进行排序

```
1. aList = ['Baidu', 'ShowMeAI', 'google']
2. print("aList : ", aList)
3.
4. aList.append( 'ByteDance' )
5. aList += ['ShowMeAI']
6. print("经过append和+计算后的aList : ", aList)
7. print("统计ShowMeAI个数 : ", aList.count('ShowMeAI'))
8.
9. aList.extend(['Taobao', 'Tencent'])
10. print("经过extend后的aList : ", aList)
11.
12. print("使用index查找Taobao的索引位置: ", aList.index('Taobao'))
13.
14. aList.insert( 3, 'DiDi')
15. print("在index为3的位置insert元素后的aList : ", aList)
16.
17. print("aList pop出的元素: ", aList.pop())
18.
19. aList.remove('ShowMeAI')
20. print("aList用remove删除第1个匹配到的ShowMeAI后: ", aList)
21.
22. aList.reverse()
23. print("aList使用reverse逆序后的结果: ", aList)
24.
25. aList.sort()
26. print("aList使用sort排序后的结果: ", aList)
```

结果

```
1. aList : ['Baidu', 'ShowMeAI', 'google']
2. 经过append和+计算后的aList : ['Baidu', 'ShowMeAI', 'google', 'ByteDance', 'ShowMeAI']
3. 统计ShowMeAI个数 : 2
4. 经过extend后的aList : ['Baidu', 'ShowMeAI', 'google', 'ByteDance', 'ShowMeAI', 'Taobao', 'Tencent']
5. 使用index查找Taobao的索引位置: 5
6. 在index为3的位置insert元素后的aList : ['Baidu', 'ShowMeAI', 'google', 'DiDi', 'ByteDance', 'ShowMeAI', 'Taobao', 'Tencent']
7. aList pop出的元素: Tencent
8. aList用remove删除第1个匹配到的ShowMeAI后: ['Baidu', 'google', 'DiDi', 'ByteDance', 'ShowMeAI', 'Taobao']
9. aList使用reverse逆序后的结果: ['Taobao', 'ShowMeAI', 'ByteDance', 'DiDi', 'google', 'Baidu']
10. aList使用sort排序后的结果: ['Baidu', 'ByteDance', 'DiDi', 'ShowMeAI', 'Taobao', 'google']
```

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 / 30:21



360P



进入bilibili,一起发弹幕吐槽!

去吐槽


一键运行所有代码

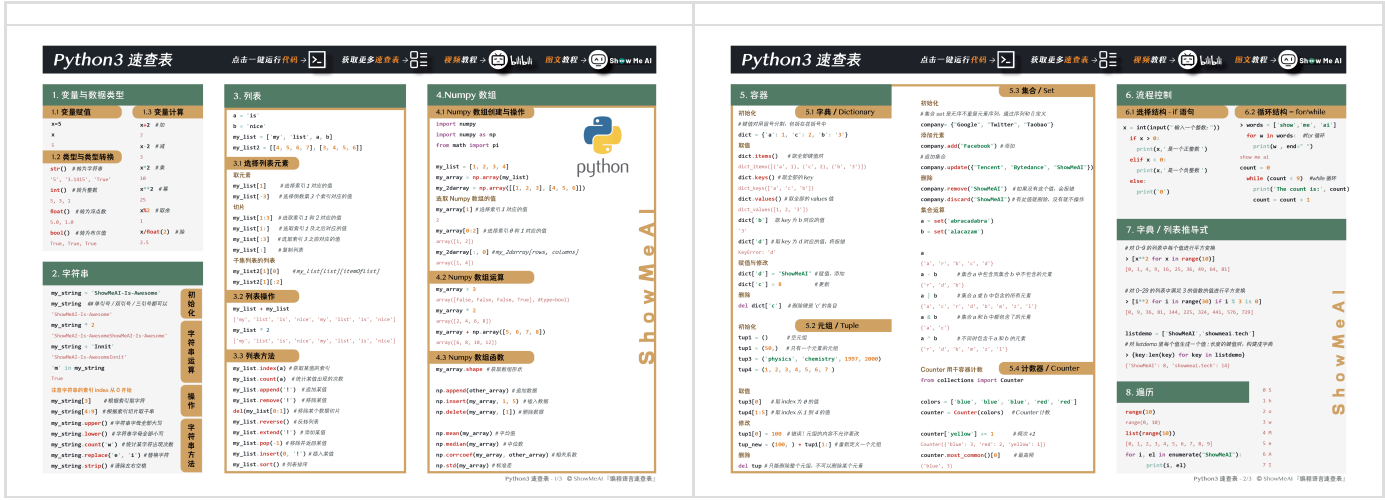
图解Python编程系列 配套的所有代码，可前往ShowMeAI 官方 **GitHub**，下载后即可在本地 Python 环境中运行。能访问 Google 的宝宝也可以直接借助 Google Colab一键运行与交互学习！

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(14) | 列表

< 上一篇

图解 Python 编程(13) | 字符串及操作

下一篇 >

图解 Python 编程(15) | 元组