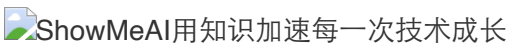


# 图解 Python 编程(4) | 基础数据类型

韩信子 2021-11-09 2089 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/67>  
声明：版权所有，转载请联系平台与作者并注明出处  
收藏ShowMeAI更多精彩内容

## Python变量类型

Python基本数据类型一般分为6种：数值（Numbers）、字符串（String）、列表（List）、元组（Tuple）、字典（Dictionary）、集合（Set）。本文详细讲解Python中变量赋值、数据类型以及数据类型的转换。

变量存储在内存中的值，这就意味着在创建变量时会在内存中开辟一个空间。基于变量的数据类型，解释器会分配指定内存，并决定什么数据可以被存储在内存中。因此，变量可以指定不同的数据类型，这些变量可以存储整数，小数或字符。

## 变量赋值

- Python 中的变量赋值不需要类型声明。
- 每个变量在内存中创建，都包括变量的标识，名称和数据这些信息。
- 每个变量在使用前都必须赋值，变量赋值以后该变量才会被创建。
- 等号 = 用来给变量赋值。

等号 = 运算符左边是一个变量名，等号 = 运算符右边是存储在变量中的值。例如（以下代码可以在在线python3环境中运行）：

```
1. num = 100 # 赋值整型变量
2. weight = 100.0 # 浮点型
3. name = "ShowMeAI" # 字符串
4.
5. print(num)
6. print(weight)
7. print(name)
```

以上实例中，100，100.0和"ShowMeAI"分别赋值给num，weight，name变量。

执行以上程序会输出如下结果：

```
1. 100
2. 100.0
3. ShowMeAI
```



## 多变量赋值

Python允许你同时为多个变量赋值。例如：

```
1. a = b = c = 1
```

以上实例，创建一个整型对象，值为1，三个变量被分配到相同的内存空间上。

您也可以为多个对象指定多个变量。例如：

```
1. a, b, c = 1, 2, "ShowMeAI"
```

以上实例，两个整型对象 1 和 2 分别分配给变量 a 和 b，字符串对象 "ShowMeAI" 分配给变量 c。

## 标准数据类型


在内存中存储的数据可以有多种类型。

例如，一个人的年龄可以用数字来存储，他的名字可以用字符来存储。

Python 定义了一些标准类型，用于存储各种类型的数据。

Python有最常用的5个标准数据类型：

- Numbers（数值）
- String（字符串）
- List（列表）
- Tuple（元组）
- Dictionary（字典）



Python常用的6种标准数据类型

Numbers（数值）

int有符号整型

float浮点型

bool布尔型

complex复数


String（字符串）


List（列表）

Tuple（元组）

Dictionary（字典）

Set（集合）

 搜索 | 微信 ShowMeAI 研究中心

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

## Python 数值

数值数据类型用于存储数值。

他们是不可改变的数据类型，这意味着改变数字数据类型会分配一个新的对象。

当你指定一个值时，Number 对象就会被创建：

```
1. num1 = 1
2. num2 = 10
```

也可以使用del语句删除一些对象的引用。

del语句的语法是：

```
1. del num1[,num2[,num3[....,numN]]]
```

可以通过使用del语句删除单个或多个对象的引用。例如：


```
1. del num
2. del num_a, num_b
```

Python支持四种不同的数字类型：

- int（有符号整型）
- float（浮点型）
- complex（复数）

一些数值类型的实例：

int	float	complex
10	0.0	3.14j
100	15.20	45.j
-786	-21.9	9.322e-36j
080	32.3e+18	.876j
-0490	-90.	-.6545+0J
-0x260	-32.54e100	3e+26J
0x69	70.2E-12	4.53e-7j




Python数据类型 | 数值（Numbers）


有符号整型 int

浮点型 float

复数 complex

10	100	-786	080	-0490	-0x260	0x69
0.0	15.20	-21.9	32.3e+18	-90.	-32.54e100	70.2E-12
3.14j	45.j	9.322e-36j	.876j	-.6545+0J	3e+26J	4.53e-7j

 搜索 | 微信 ShowMeAI 研究中心

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

## Python字符串

字符串或串(String)是由数字、字母、下划线组成的一串字符。

一般记为：

```
1. s = "a1a2...an" # n>=0
```

它是编程语言中表示文本的数据类型。

python的字符串列表有2种取值顺序：

- 从左到右索引默认0开始的，最大范围是字符串长度少1
- 从右到左索引默认-1开始的，最大范围是字符串开头

从后面索引

从前面索引

从前面截取

从后面截取

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

01234567

|s|h|o|w|M|e|A|I|

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

:1234567:

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

搜索 | 微信 ShowMeAI 研究中心

http://www.showmeai.tech/

如果你要实现从字符串中获取一段子字符串的话，可以使用【头下标:尾下标】来截取相应的字符串，其中下标是从 0 开始算起，可以是正数或负数，下标可以为空表示取到头或尾。

【头下标:尾下标】获取的子字符串包含头下标的字符，但不包含尾下标的字符。

比如：

```
1. >>> s = 'ShowMeAI'
2. >>> s[6:8]
3. 'AI'
```

当使用以冒号分隔的字符串，python 返回一个新的对象，结果包含了以这对偏移标识的连续的内容，左边的开始是包含了下边界。

上面的结果包含了 **s[1]** 的值 **b**，而取到的最大范围不包括尾下标，就是 **s[5]** 的值 **f**。

可以使用加号 (+) 对字符串进行连接，使用星号 (\*) 对字符串进行重复操作。如下（以下代码可以在在线python3环境中运行）：

```
1. str = 'Hello ShowMeAI!'
2.
3. print(str) # 输出完整字符串
4. print(str[0]) # 输出字符串中的第一个字符
5. print(str[2:5]) # 输出字符串中第三个至第六个之间的字符串
6. print(str[2:]) # 输出从第三个字符开始的字符串
7. print(str * 2) # 输出字符串两次
8. print(str + " Awesome") # 输出连接的字符串
```

以上实例输出结果：

```
1. Hello ShowMeAI!
2. H
3. llo
4. llo ShowMeAI!
5. Hello ShowMeAI!Hello ShowMeAI!
6. Hello ShowMeAI! Awesome
```

Python 列表截取可以接收第三个参数，参数作用是截取的步长，以下实例在索引 1 到索引 4 的位置并设置为步长为 2（间隔一个位置）来截取字符串：

Python数据类型 | 字符串 (String)

01234567

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

>>> letters[1:4:2]

['h', 'w']

2

> Python字符串详解

http://showmeai.tech/article-detail/python-string

搜索 | 微信 ShowMeAI 研究中心

http://www.showmeai.tech/

更多python字符串的详细讲解知识可以参考python字符串及操作

## Pvthon列表

https://www.showmeai.tech/article-detail/67

3/7

Python 数据类型

List（列表） 是 Python 中使用最频繁的数据类型。

列表可以完成大多数集合类的数据结构实现。它支持字符，数字，字符串甚至可以包含列表（即嵌套）。

列表用 [] 标识，是 python 最通用的复合数据类型。

列表中值的切割也可以用到变量 [头下标:尾下标]，就可以截取相应的列表，从左到右索引默认 0 开始，从右到左索引默认 -1 开始，下标可以为空表示取到头或尾。

Python数据类型 | 列表 (List)

```
>>> t = ['S', 'h', 'o', 'w', 'M', 'e', 'A', 'I']
```

索引

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

01234567

```
>>>t[1:3]
['S', 'h']

>>>t[:4]
['S', 'h', 'o', 'w']
```

http://www.showmeai.tech/

> Python列表详解  
http://showmeai.tech/article-detail/python-list

搜索 | 微信 ShowMeAI 研究中心

Python数据类型 | 列表 (List)

```
>>> t = ['S', 'h', 'o', 'w', 'M', 'e', 'A', 'I']
```

截取

:1234567:

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

```
>>>t[6:]
['A', 'I']

>>>t[:]
['S', 'h', 'o', 'w', 'M', 'e', 'A', 'I']
```

http://www.showmeai.tech/

搜索 | 微信 ShowMeAI 研究中心

加号 + 是列表连接运算符，星号 \* 是重复操作。如下（以下代码可以在在线python3环境中运行）：

1. list = [ 'ShowMeAI', 786 , 2.23, 'show', 70.2 ]

2. tinylist = [123, 'show']

3.

4. print(list) # 输出完整列表

5. print(list[0]) # 输出列表的第一个元素

6. print(list[1:3]) # 输出第二个至第三个元素

7. print(list[2:]) # 输出从第三个开始至列表末尾的所有元素

8. print(tinylist \* 2) # 输出列表两次

9. print(list + tinylist) # 打印组合的列表

以上实例输出结果：

1. ['ShowMeAI', 786, 2.23, 'show', 70.2]

2. ShowMeAI

3. [786, 2.23]

4. [2.23, 'show', 70.2]

5. [123, 'show', 123, 'show']

6. ['ShowMeAI', 786, 2.23, 'show', 70.2, 123, 'show']

更多python列表的详细讲解知识可以参考python列表

Python元组

元组是另一个数据类型，类似于 List（列表）。

Python数据类型 | 元组 (Tuple)

```
>>> tup1 = ( 'google' , 'ShowMeAI' , 2021 , 2022 , 123.4 , 0 )
```

-6-5-4-3-2-1

012345

googleShowMeAI20212022123.40

反向索引 ← -6 -1

正向索引 0 5 →

http://www.showmeai.tech/

> Python元组详解  
http://showmeai.tech/article-detail/python-tuple

搜索 | 微信 ShowMeAI 研究中心

元组用 () 标识。内部元素用逗号隔开。但是元组不能二次赋值，相当于只读列表。（以下代码可以在在线python3环境中运行）

1. tuple = ( 'ShowMeAI', 786 , 2.23, 'show', 70.2 )

https://www.showmeai.tech/article-detail/67

4/7



```
2. tinytuple = (123, 'show')
3.
4. print(tuple) # 输出完整元组
5. print(tuple[0]) # 输出元组的第一个元素
6. print(tuple[1:3]) # 输出第二个至第四个（不包含）的元素
7. print(tuple[2:]) # 输出从第三个开始至列表末尾的所有元素
8. print(tinytuple * 2) # 输出元组两次
9. print(tuple + tinytuple) # 打印组合的元组
```

以上实例输出结果：

```
1. ('ShowMeAI', 786, 2.23, 'show', 70.2)
2. ShowMeAI
3. (786, 2.23)
4. (2.23, 'show', 70.2)
5. (123, 'show', 123, 'show')
6. ('ShowMeAI', 786, 2.23, 'show', 70.2, 123, 'show')
```

以下是元组无效的，因为元组是不允许更新的。而列表是允许更新的：


```
1. tuple = ( 'ShowMeAI', 345 , 2.23, 'show', 456.2 )
2. list = [ 'ShowMeAI', 345 , 2.23, 'show', 456.2 ]
3. tuple[2] = 100    # 元组中是非法应用
4. list[2] = 100     # 列表中是合法应用
```

更多python元组的详细讲解知识可以参考[python元组](#)

## Python字典

字典(dictionary)是除列表以外python之中最灵活的内置数据结构类型。列表是有序的对象集合，字典是无序的对象集合。

两者之间的区别在于：字典当中的元素是通过键来存取的，而不是通过偏移存取。



### Python数据类型 | 字典 (Dictionary)

```
>>> d = { key1 : value1 , key2 : value2 , key3 : value3 }
```

字典

key1

key2

key3

value1

value2

value3

> Python字典详解

<http://showmeai.tech/article-detail/python-dictionary>

搜索 | 微信 ShowMeAI 研究中心

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

字典用“{ }”标识。字典由索引(key)和它对应的值value组成。（以下代码可以在在线python3环境中运行）

```
1. dict = {}
2. dict['one'] = "This is one"
3. dict[2] = "This is two"
4.
5. tinydict = {'name': 'ShowMeAI','code':3456, 'dept': 'AI'}
6.
7. print(dict['one']) # 输出键为'one' 的值
8. print(dict[2]) # 输出键为 2 的值
9. print(tinydict) # 输出完整的字典
10. print(tinydict.keys()) # 输出所有键
11. print(tinydict.values()) # 输出所有值
```

输出结果为：

```
1. This is one
2. This is two
3. {'name': 'ShowMeAI', 'code': 3456, 'dept': 'AI'}
4. dict_keys(['name', 'code', 'dept'])
5. dict_values(['ShowMeAI', 3456, 'AI'])
```

更多python字典的详细讲解知识可以参考[python字典](#)

## Python数据类型转换

有时候，我们需要对数据内置的类型进行转换，数据类型的转换，你只需要将数据类型作为函数名即可。

以下几个内置的函数可以执行数据类型之间的转换。这些函数返回一个新的对象，表示转换的值。

函数	描述
int(x [,base])	将x转换为一个整数
long(x [,base] )	将x转换为一个长整数
float(x)	将x转换到一个浮点数
complex(real [,imag])	创建一个复数

图解 Python 编程(4)   基础数据类型	
<code>complex(real [,imag])</code>	创建一个复数
<code>str(x)</code>	将对象 x 转换为字符串
<code>repr(x)</code>	将对象 x 转换为表达式字符串
<code>eval(str)</code>	用来计算在字符串中的有效Python表达式,并返回一个对象
<code>tuple(s)</code>	将序列 s 转换为一个元组
<code>list(s)</code>	将序列 s 转换为一个列表
<code>set(s)</code>	转换为可变集合
<code>dict(d)</code>	创建一个字典。d 必须是一个序列 (key,value)元组。
<code>frozenset(s)</code>	转换为不可变集合
<code>chr(x)</code>	将一个整数转换为一个字符
<code>unichr(x)</code>	将一个整数转换为Unicode字符
<code>ord(x)</code>	将一个字符转换为它的整数值
<code>hex(x)</code>	将一个整数转换为一个十六进制字符串
<code>oct(x)</code>	将一个整数转换为一个八进制字符串

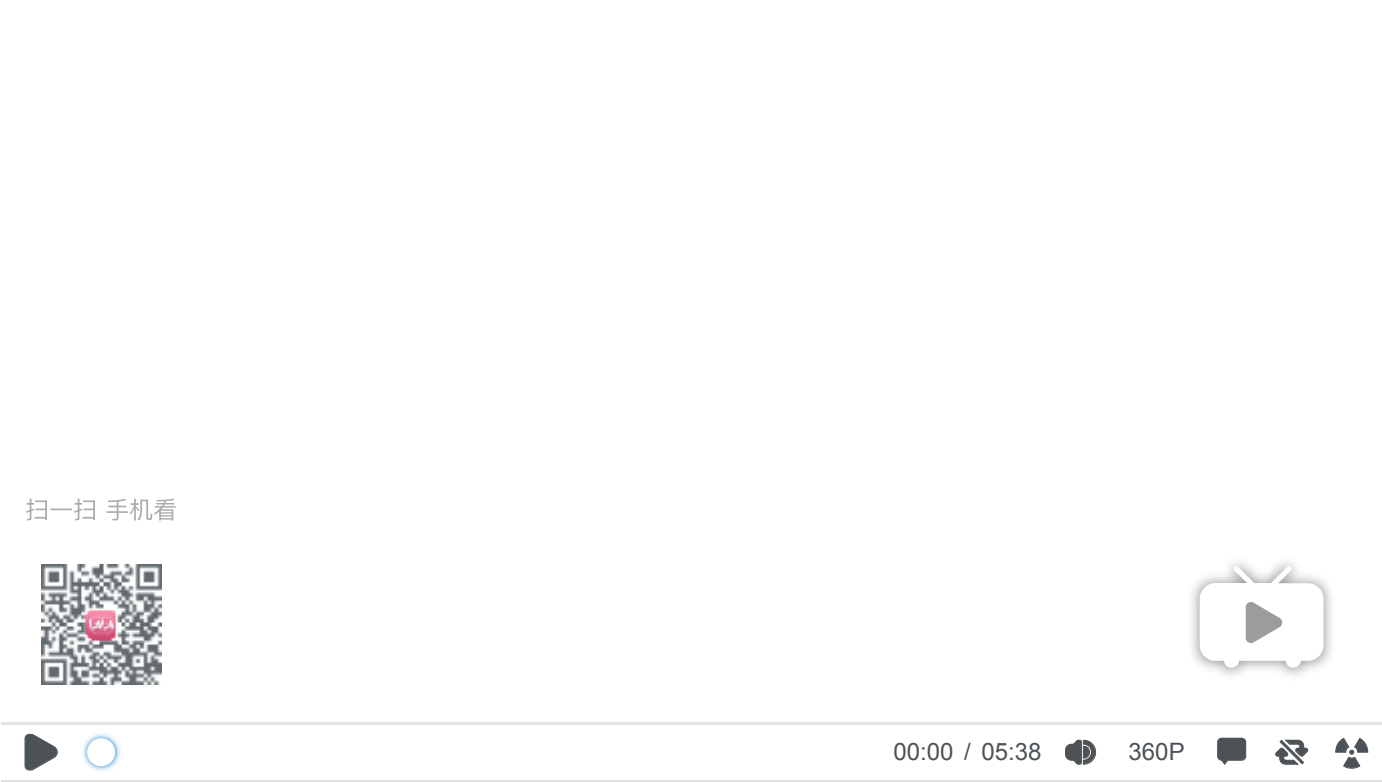
视频教程

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



【双语字幕+资料下载】Python 3全系列...

去bilibili观看

分享



【双语字幕+资料下载】Python 3全系列...

去bilibili观看

分享

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！

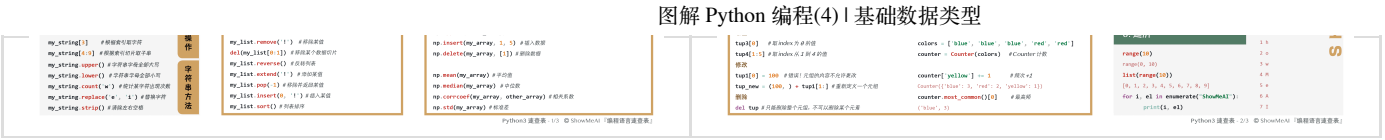
下载Python要点速查表

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(4) | 基础数据类型

[< 上一篇](#)

图解 Python 编程(3) | 基础语法

[下一篇 >](#)

图解 Python 编程(5) | 运算符