



# python 魔鬼训练营 第2周

DATAGURU专业数据分析社区

**【声明】** 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>

## 第2课：Python数据结构

课程内容：

- 元组
- 列表
- 字典
- 集合

本次课程内容重点讲解Python的常用数据结构的一些用法，我们平常用到的可能都是最常用的几个方法，比如：切片，增删等；这里会更加多的讲解些平常不经常用到的方法和场景，让我们更加熟悉和了解Python数据结构的特点和特性。

# 第1节：Python元组

## ■ 元组的结构

T = tuple()

T = tuple([1])

T = ()

T = (1,)

T = (1,2,3,4)

T = (0, None, False, " ")

T = (1, True, 'hello world' , [1,2])

# 第1节：Python元组

## ■ 元组的成员

>dir(tuple)

```
['_add_', '__class__', '__contains__', '__delattr__', '__doc__', '__eq__', '__format__', '__ge__',  
 '__getattr__', '__getitem__', '__getnewargs__', '__getslice__', '__gt__', '__hash__',  
 '__init__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__',  
 '__reduce_ex__', '__repr__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__',  
 'count', 'index']
```

# 第1节：Python元组

## ■ 元组的方法

Count():

`(1,2,3,4,1).count(1) => 2`

Index():

`(1,2,3,4).index(2) => 1`

`(1,2,3,4,1).index(1) => ?`

# 第1节：Python元组

## ■ 元组的应用

```
a = (1,2,3,4)
```

```
a[0] => 1
```

```
a[2:] => (3,4)
```

有序不可变的tuple有什么意义？

因为tuple不可变，所以代码更安全；内容确保不会在任意环节中被改变；通常是作为参数在函数调用时被使用，保证内容没有被所调用的函数所改变。

## 第2节：Python列表

### ■ 列表的结构

```
l = list()
```

```
l = list((1,2))
```

```
l = []
```

```
l = [1]
```

```
l = [1,2,3,4]
```

```
l = [0, None, False, ""]
```

```
l = [1, True, 'hello world', (1,2)]
```



## 第2节：Python列表

### ■ 列表的成员

> dir(list)

```
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__delslice__', '__doc__',  
 '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getslice__', '__gt__',  
 '__hash__', '__iadd__', '__imul__', '__init__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',  
 '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__',  
 '__setattr__', '__setitem__', '__setslice__', '__sizeof__', '__str__', '__subclasshook__', 'append',  
 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

## 第2节：Python列表

### ■ 列表的方法

'count' , 'index ' => 同 tuple

'append ' => 追加

'extend ' => 扩展

'insert ' => 插入

'pop ' => 弹出

'remove ' => 删除

'reverse ' => 反转

'sort ' => 排序

## 第2节：Python列表

### ■ 列表的应用

```
l = [1,2,3,4]
```

```
l[0] => 1
```

```
l[2:] => (3,4)
```

有序可变的list有什么意义？

可变是为了可以保存过程的中间结果，动态的保存了函数体内动态执行结果。

## 第3节：Python字典

### ■ 字典的结构

```
d = dict()
```

```
d = dict(a=2, c=4)
```

```
d = dict([(1,2), (3,4)])
```

```
d = dict({1:2, 3:4})
```

```
d = { 'a' : 1, 'b' : 2, 'c' : (1,2), 'd' : [None, True]}
```

### ■ 字典的成员

> dir(dict)

```
['__class__', '__cmp__', '__contains__', '__delattr__', '__delitem__', '__doc__', '__eq__',  
 '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__init__',  
 '__iter__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_ex__',  
 '__repr__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'clear',  
 'copy', 'fromkeys', 'get', 'has_key', 'items', 'iteritems', 'iterkeys', 'itervalues', 'keys', 'pop',  
 'popitem', 'setdefault', 'update', 'values', 'viewitems', 'viewkeys', 'viewvalues']
```

## 第3节：Python字典

### ■ 字典的方法

'clear',

'copy',

'fromkeys',

'get',

'has\_key',

'items',

'iteritems',

'iterkeys',

## 第3节：Python字典

### ■ 字典的方法

'itervalues',

'keys',

'pop',

'popitem',

'setdefault',

'update',

'values',

'viewitems',

## 第3节：Python字典

### ■ 字典的方法

'viewkeys',

'viewvalues'



## 第3节：Python字典

### ■ 字典的应用

```
d = {1:2, 3:4}
```

```
d[1] => 2
```

```
d[3] = 5
```

```
d[3] => 5
```

无序可变的dict有什么意义？

映射key-value对通常用来作为hash存储。

## 第4节：Python集合

### ■ 集合的结构

```
s = set()
```

```
s = set( 'hello' ) => set([ 'h' , 'e' , 'l' , 'o' ])
```

```
s = set([1,2,3,4]) => set([1,2,3,4])
```

```
s = set((1,2,3,4)) => set([1,2,3,4])
```

## 第4节：Python集合

### ■ 集合的成员

```
> dir(set)
```

```
['_and_', '__class__', '__cmp__', '__contains__', '__delattr__', '__doc__', '__eq__', '__format__',  
 '__ge__', '__getattr__', '__gt__', '__hash__', '__iand__', '__init__', '__ior__', '__isub__',  
 '__iter__', '__ixor__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__or__', '__rand__',  
 '__reduce__', '__reduce_ex__', '__repr__', '__ror__', '__rsub__', '__rxor__', '__setattr__', '__sizeo  
f__', '__str__', '__sub__', '__subclasshook__', '__xor__', 'add', 'clear', 'copy', 'difference',  
 'difference_update', 'discard', 'intersection', 'intersection_update', 'isdisjoint', 'issubset',  
 'issuperset', 'pop', 'remove', 'symmetric_difference', 'symmetric_difference_update',  
 'union', 'update']
```

### ■ 集合的方法

'add', 'clear', 'copy', 'difference', 'difference\_update' => -=, 'discard', 'intersection',  
'intersection\_update' => &=, 'isdisjoint', 'issubset', 'issuperset', 'pop', 'remove',  
'symmetric\_difference', 'symmetric\_difference\_update' => ^=, 'union', 'update' => |=

非运算符版本的 update(), intersection\_update(), difference\_update()和 symmetric\_difference\_update()将会接受任意 iterable 作为参数

## 第4节：Python集合

### ■ 集合的应用

```
s = set([3,5,9,10])
```

```
t = set("Hello")
```

- $a = t \mid s$       #  $t$  和  $s$  的并集
- $b = t \& s$       #  $t$  和  $s$  的交集
- $c = t - s$       # 求差集（项在 $t$ 中，但不在 $s$ 中）
- $d = t \wedge s$       # 对称差集（项在 $t$ 或 $s$ 中，但不会同时出现在二者中）

## 第5节：浅复制和深复制

### ■ 它们的区别

浅复制：只复制父对象，不会拷贝对象的内部的子对象

深复制：拷贝对象及其子对象

Import copy

copy.copy() ##浅复制

copy.deepcopy() ##深复制

## 第5节：浅复制和深复制

### ■ 它们的应用

```
import copy
```

```
a = [1, 2, 3, 4, ['a', 'b']] #原始对象
```

```
b = a #赋值，传对象的引用
```

```
c = copy.copy(a) #对象拷贝，浅拷贝
```

```
d = copy.deepcopy(a) #对象拷贝，深拷贝
```

```
a.append(5) #修改对象a
```

```
a[4].append('c') #修改对象a中的['a', 'b']数组对象
```

## 第5节：浅复制和深复制

```
print 'a = ', a
```

```
print 'b = ', b
```

```
print 'c = ', c
```

```
print 'd = ', d
```

```
==>>
```

```
a = [1, 2, 3, 4, [ 'a' , 'b' , 'c' ], 5] ##原对象
```

```
b = [1, 2, 3, 4, [ 'a' , 'b' , 'c' ], 5] # 引用对象
```

```
c = [1, 2, 3, 4, [ 'a' , 'b' , 'c' ]] #浅复制
```

```
d = [1, 2, 3, 4, [ 'a' , 'b' ]] #深复制
```



## 第2课：Python数据结构

要点回顾：

- 元组的特点及用法
- 列表的特点及用法
- 字典的特点及用法
- 集合的特点及用法

- Dataguru ( 炼数成金 ) 是专业数据分析网站 , 提供教育 , 媒体 , 内容 , 社区 , 出版 , 数据分析业务等服务。我们的课程采用新兴的互联网教育形式 , 独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围 , 重竞争压力的特点 , 同时又发挥互联网的威力打破时空限制 , 把天南地北志同道合的朋友组织在一起交流学习 , 使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本 , 直线下降至百元范围 , 造福大众。我们的目标是 : 低成本传播高价值知识 , 构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情 , 请看我们的培训网站 <http://edu.dataguru.cn>

# Thanks

**FAQ时间**