

元组

- 目标
 - 元组的应用场景
 - 定义元组
 - 元组常见操作

1、元组的应用场景

思考：如果想要存储多个数据，但是这些数据是不能修改的数据，怎么做？

答：列表？列表可以一次性存储多个数据，但是列表中的数据允许更改。

```
num_list = [10, 20, 30]
num_list[0] = 100
```

一个元组可以存储多个数据，元组内的数据是不能修改的

2、定义元组

元组特点：定义元组使用 小括号，且 逗号 隔开各个数据，数据可以是不同的数据类型。

```
# 多个数据元组
t1 = (10, 20, 30)

# 单个数据元组
t2 = (10,)
```

注意：如果定义的元组只有一个数据，那么这个数据后面也好添加逗号，否则数据类型为唯一的这个数据的数据类型

```
t2 = (10,)
print(type(t2)) # tuple

t3 = (20)
print(type(t3)) # int

t4 = ('hello')
print(type(t4)) # str
```

3、元组的常见操作

元组数据不支持修改，只支持查找，具体如下：

- 按下标查找数据

```
tuple1 = ('aa', 'bb', 'cc', 'bb')
print(tuple1[0]) # aa
```

- index(): 查找某个数据, 如果数据存在返回对应的下标, 否则报错, 语法和列表、字符串的index方法相同。

```
tuple1 = ('aa', 'bb', 'cc', 'bb')
print(tuple1.index('aa')) # 0
```

- count(): 统计某个数据在当前元组出现的次数。

```
tuple1 = ('aa', 'bb', 'cc', 'bb')
print(tuple1.count('bb')) # 2
```

- len(): 统计元组中数据的个数。

```
tuple1 = ('aa', 'bb', 'cc', 'bb')
print(len(tuple1)) # 4
```

注意: 元组内的直接数据如果修改则立即报错

```
tuple1 = ('aa', 'bb', 'cc', 'bb')
tuple1[0] = 'aaa'
```

但是如果元组里面有列表, 修改列表里面的数据则是支持的

```
tuple2 = (10, 20, ['aa', 'bb', 'cc'], 50, 30)
print(tuple2[2]) # 访问到列表

# 结果: (10, 20, ['aaaaa', 'bb', 'cc'], 50, 30)
tuple2[2][0] = 'aaaaa'
print(tuple2)

# 结果: (10, 20, ['aaaaa', 'bb', 'cc'], 50, 30)
```

4、总结

- 定义元组

```
t1 = (10, 20, 30)
t2 = (10,) # 定义只有一个数据的元组
```

- 常用操作方法

- `index()`
- `len()`