

课程：元组

目标

- 元组的应用场景
- 定义元组
- 元组常见操作

一. 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。

```
1 num_list = [10, 20, 30]
2 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。

二. 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。

```
1 # 多个数据元组
2 t1 = (10, 20, 30)
3
4 # 单个数据元组
5 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

```
1 t2 = (10,)
2 print(type(t2)) # tuple
3
4 t3 = (20)
5 print(type(t3)) # int
6
7 t4 = ('hello')
8 print(type(t4)) # str
```

三. 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

```
1 tuple1 = ('aa', 'bb', 'cc', 'bb')
2 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。

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

- count(): 统计某个数据在当前元组出现的次数。

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

- len(): 统计元组中数据的个数。

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

注意：元组内的直接数据如果修改则立即报错

```
1 tuple1 = ('aa', 'bb', 'cc', 'bb')
2 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，故自觉很重要。

```
1 tuple2 = (10, 20, ['aa', 'bb', 'cc'], 50, 30)
2 print(tuple2[2]) # 访问到列表
3
4 # 结果: (10, 20, ['aaaaa', 'bb', 'cc'], 50, 30)
5 tuple2[2][0] = 'aaaaa'
6 print(tuple2)
```

四. 总结

- 定义元组

```
1 t1 = (10, 20, 30)
2
3 t2 = (10,)
```

- 常用操作方法
 - index()
 - len()