



# 九章算法基础班

## 第三讲 线性数据结构 I

课程版本 : v2.0 张三疯 老师



扫描二维码关注微信/微博  
获取最新面试题及权威解答

微信: [ninechapter](#)

知乎专栏 : <http://zhuanlan.zhihu.com/jiuzhang>

微博: <http://www.weibo.com/ninechapter>

官网: [www.jiuzhang.com](http://www.jiuzhang.com)

九章课程不提供视频，也严禁录制视频的侵权行为  
否则将追求法律责任和经济赔偿  
请不要缺课

# 本节重点

---

- 列表 ( List ) 和元组 ( Tuple )
- 字符串 ( String )
- 引用 ( Reference )

# 课程回顾

- 顺序结构
  - 从上到下顺序执行语句
- 控制结构
- 循环结构

# 循环结构

- **for 循环**

```
1 scores = [90, 95, 92, 89, 100, 98]  
2  
3 for score in scores:  
4     print(score)                      循环变量  
5  
6 for i in range(len(scores)):  
7     print(scores[i])
```

# 循环结构

- 如何灵活遍历指定位置的值呢？

```
range(n) # [0, 1, ..., n - 1]
```

```
range(m, n) # [m, m + 1, ..., n - 1]
```

```
range(m, n, k) # [m, m + k, m + 2k, ...]
```

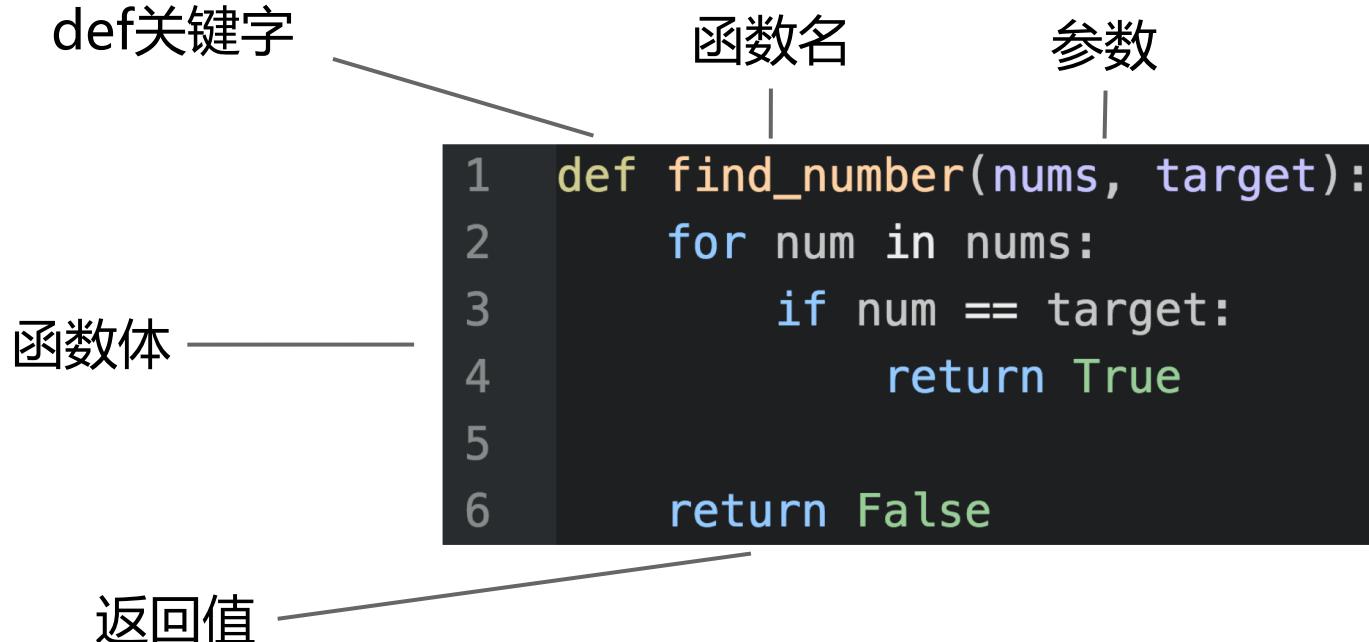
```
range(n, m, -1) # [n, n - 1, n - 2, ..., m + 1]
```

- enumerate函数

- 可以同时获取下标和值

```
for (i, score) in enumerate(nums):  
    print(i, score)
```

# 函数 Function



# 类和对象

- 什么是面向对象 ( Object - Oriented ) ?
  - 面向对象是一种世界观：世间万物皆为对象
  - 面向对象是一种程序设计方式



# 类和对象

- 在Python中，类是对现实事物的抽象

```
1 class Student():
2
3     def __init__(self, name, score):
4         self.name = name
5         self.score = score
6
7     def speak(self):
8         print(self.name, self.score)
```

class关键字

类名 ( upper camel case )

方法 ( 行为 )

变量 ( 属性 )

# 类和对象

- 创建对象

```
1 student = Student('Jack', 80)
2 student.score = 98
3 print(student.name, student.score)
4 student.speak()
```

- 什么是实例 ( instance ) ?
  - 实例就是对象

# 列表 ( List ) 和元组 ( Tuple )

- 什么是数据结构 ( data structure )

- 数据，结构，操作
- 线性数据结构

- 操作

- CRUD
- 增查改删



# 列表 (List)

- Python的基本数据结构之一
  - 任意对象的**有序**集合
  - list中的元素不一定是同一类型，非常灵活

```
list_1 = [12, 15.6, True, 'hello', ['a', 'b']]  
list_2 = [1, 2, 3, 4]  
list_3 = list('hello') # ['h', 'e', 'l', 'l', 'o']
```

# 列表 (List)

- list的常见操作
  - 增 ( Create ) : +, \*, append, insert, extend
  - 查 ( Read ) : 迭代 ( iteration ), 索引, 切片 ( slice ), in, index, count
  - 改 ( Update ) : 索引赋值 , 切片赋值
  - 删 ( Delete ) : pop , remove , del
  - 其他 : len, sort, reverse

- 练习一：面试真题
  - Remove Element
  - <http://www.lintcode.com/en/problem/remove-element/>
  - <https://www.jiuzhang.com/solution/remove-element/>

# 元组 (Tuple)

- Python的基本数据结构之一
  - 任意对象的有序集合
  - tuple中的元素不一定是同一类型，非常灵活
  - 固定长度，元素不可变（immutable）！！！

```
tuple_1 = (12, 15.6, True, 'hello', ['a', 'b'])
tuple_2 = 1, 2, 3, 4
tuple_3 = tuple('hello') # ('h', 'e', 'l', 'l', 'o')
tuple_4 = (1,)
```

# 元组 (Tuple)

- tuple的操作
  - 对比list，tuple没有修改自身元素的操作
  - 任何对于tuple的修改都会报错
- 思考：tuple存在的意义？

# 列表（List）和元组（Tuple）



- 互相转化
  - `tuple([1, 2, 3])` → `(1, 2, 3)`
  - `list((1, 2, 3))` → `[1, 2, 3]`

# 字符串 String

# 字符串 (String)

- Python的基本数据结构之一
  - 字符的有序集合
  - 固定长度，不可变 ( immutable ) !!!
  - 可以使用单引号或者双引号，在代码中保持统一

```
str_1 = 'Hello world!'
str_2 = 'Jiuzhang'
str_3 = "spam's"
```

- Python中不存在字符类型，只有单个字符的字符串
- 普通字符 & 转义字符
- 字符的存储
  - 以整数的形式存储
  - `ord()` & `chr()`

# 字符 (Character)

---



- ASCII表 ( 0 - 127 )
  - <http://tool.oschina.net/commons?type=4>
- Unicode表 ( 0 - 65535 )
  - <https://unicode-table.com/en/#control-character>

# 字符 (Character)

- 练习二：将一个给定的小写字母转化为对应的大写字母

- 解法一：

```
lower_char = 'm'  
upper_char = chr(ord(lower_char) - ord('a') + ord('A'))
```

- 解法二：

```
upper_char = lower_char.upper()
```

# 字符串 (String)

- string的常见操作
  - 连接 ( +, \* )
  - 迭代 ( iteration ), 索引, 切片 ( slice ), find, replace
  - 格式化字符串
  - 长度 ( len )

# 字符串 (String)

- string与其他类型的转换
  - 连接字符串时要特别注意

```
print('My score is' + str(100))
float_str = str(10.5)
float_num = float(float_str)
```

- 注意：字符串本身不可修改

# 字符串 (String)

- 练习三：面试真题
  - String to Integer
  - <http://www.lintcode.com/en/problem/string-to-integer/>
  - <https://www.jiuzhang.com/solution/string-to-integer/>

# 引用 Reference

# 引用 (Reference)

- 对象模型
  - Example 1

```
20 student_1 = Student('Tom')
21 student_2 = Student('Jack')
```

name : 'Tom'  
score : 100  
speak()

student\_1

name : 'Jack'  
score : 100  
speak()

student\_2

# 引用 (Reference)

- 对象模型
  - Example 1

name : 'Tom'  
score : 100  
speak()

student\_1

name : 'Jerry'  
score : 100  
speak()

student\_2

```
25 student_2.name = 'Jerry'  
26 student_1.speak()  
27 student_2.speak()
```

# 引用 (Reference)

- 对象模型
  - Example 2

```
32 student_1 = Student('Tom')
33 student_2 = student_1
```

name : 'Tom'  
score : 100  
speak()

student\_1

name : 'Tom'  
score : 100  
speak()

student\_2

?

# 引用 (Reference)

- 对象模型
  - 原样复制

```
32 student_1 = Student('Tom')  
33 student_2 = student_1
```

name : 'Tom'  
score : 100  
speak()

student\_1

name : 'Tom'  
score : 100  
speak()

student\_2

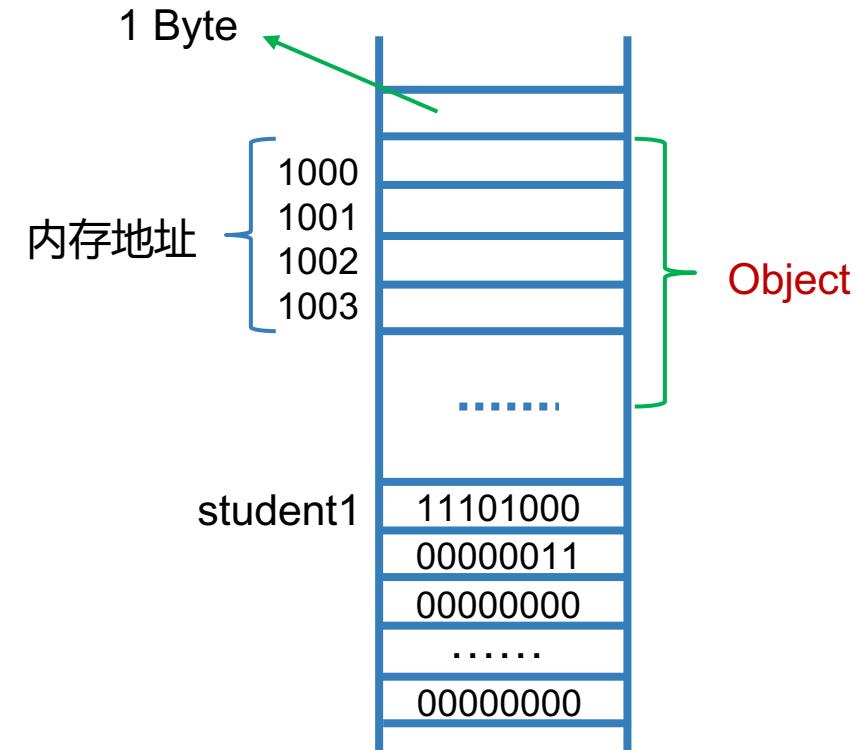


# 引用 (Reference)

- 内存模型
  - 引用



```
32 student_1 = Student('Tom')
```



# 引用 (Reference)

- 内存模型
  - 引用赋值

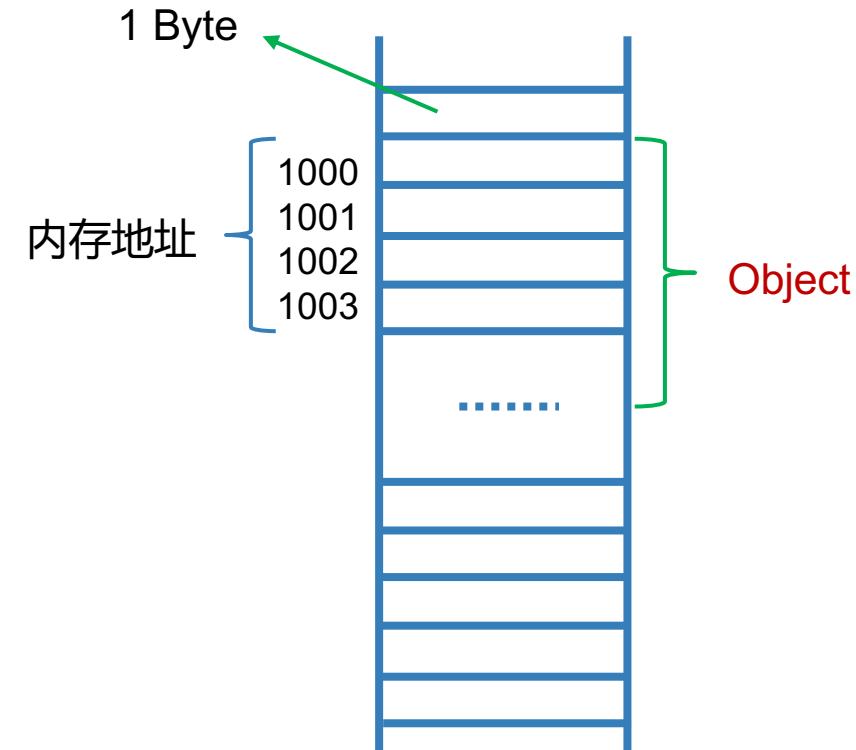
1000

student\_1

1000

student\_2

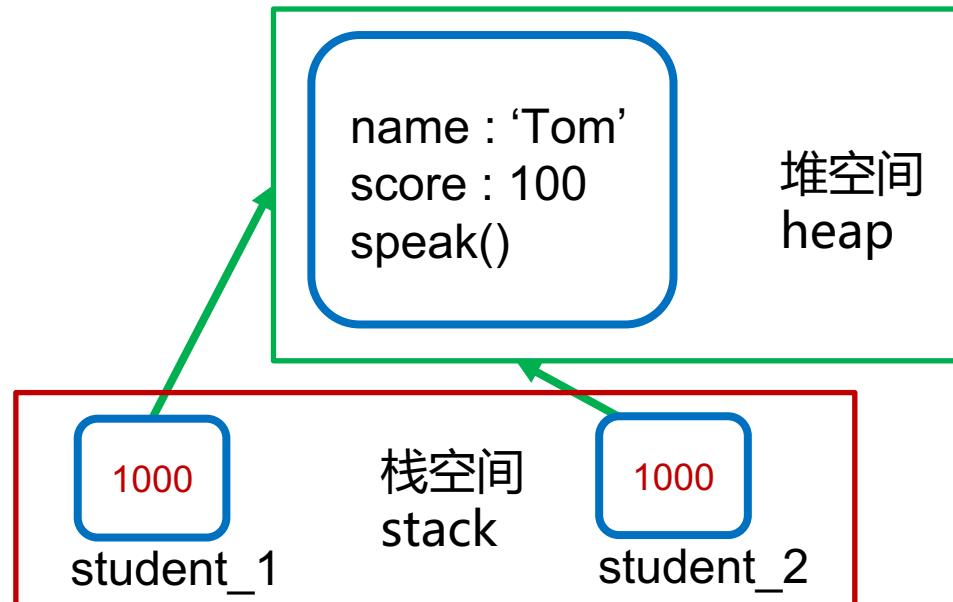
33 student\_2 = student\_1



# 引用 (Reference)

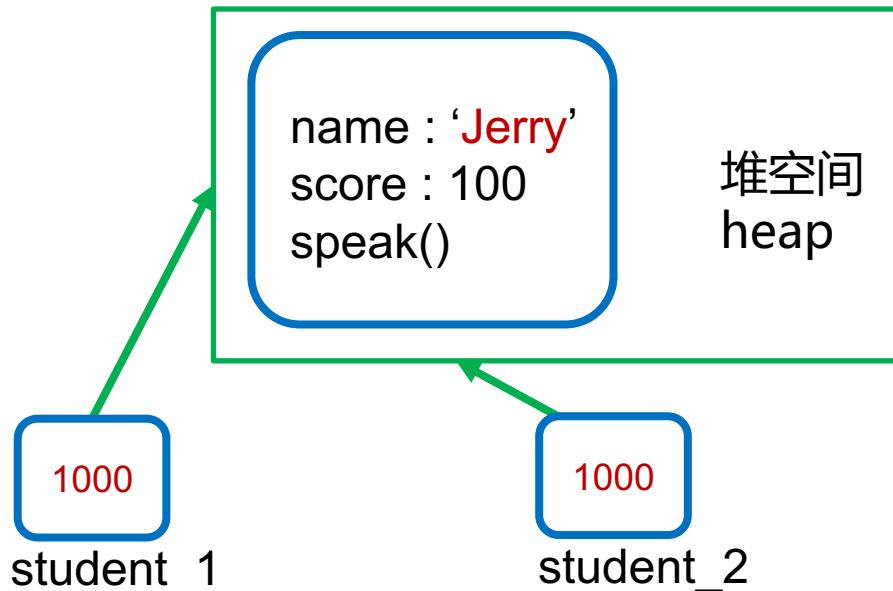
- 对象模型
  - 引用

```
32 student_1 = Student('Tom')
33 student_2 = student_1
```



# 引用 (Reference)

- 对象模型
  - 引用

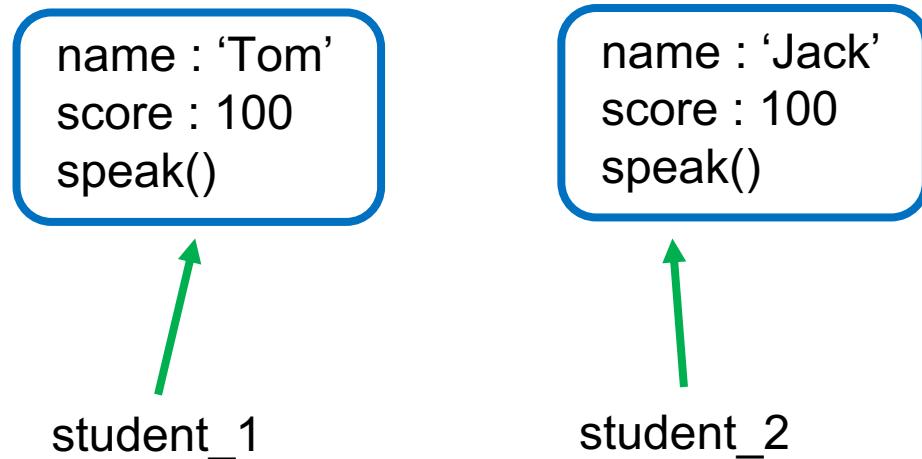


```
37 student_2.name = 'Jerry'  
38 student_1.speak()  
39 student_2.speak()
```

# 引用 (Reference)

- 对象模型
  - Example 3

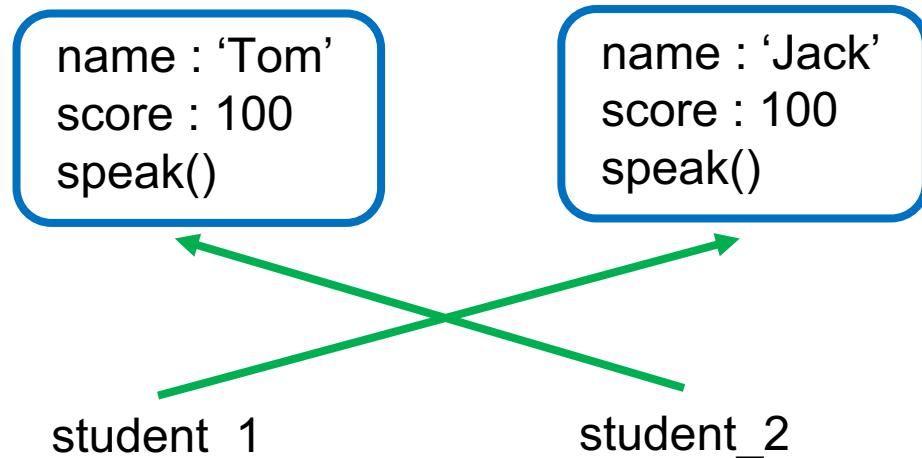
```
44 student_1 = Student('Tom')  
45 student_2 = Student('Jack')
```



# 引用 (Reference)

- 对象模型
  - Example 3

```
49 t = student_1
50 student_1 = student_2
51 student_2 = t
```



# 引用 (Reference)

- 什么是引用 ( Reference ) ?
  - 引用好比遥控器，对象好比电视机
- Python中所有变量都是引用
  - 存储的是对象的地址
  - id, is

# 引用 (Reference)

- 赋值操作和函数传参都是复制地址 ( Copy address )
- 修改引用 vs. 修改对象
  - list和tuple内部存储的也都是引用

```
my_list = [20, 16, 34, 51, 66]
my_list = my_list[1:4]
my_list[2] = 11
```



# 总结

---

- 数据结构
- 列表 ( List ) 和元组 ( Tuple )
  - 常见操作，注意区别
- 字符串 ( String )
  - 面试重点
- 引用 ( Reference )



扫描二维码关注微信/微博  
获取最新面试题及权威解答

微信: [ninechapter](#)

微博: <http://www.weibo.com/ninechapter>

官网: [www.jiuzhang.com](http://www.jiuzhang.com)



谢谢大家