



九章算法基础班

第二讲 循环、函数和面向对象

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

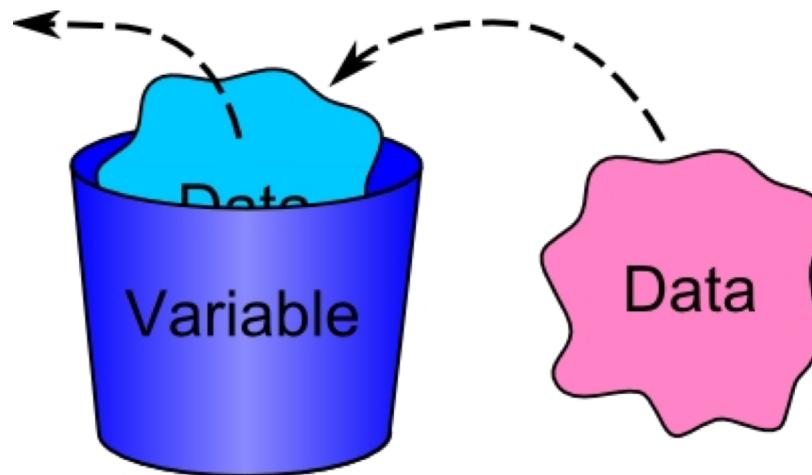
九章课程不提供视频，也严禁录制视频的侵权行为
否则将追求法律责任和经济赔偿
请不要缺课

本节重点

- 程序控制流 (Control flow)
- 函数 (Function)
- 面向对象入门 (Object Oriented)

课程回顾

- 变量 (variable) : 存储数据的容器



变量及运算

- 整数 (int) 及其运算
 - 数学运算 : +, -, *, / , //, %
 - 赋值运算 : =, += , -=, *=, /=, //=, %=

- 浮点数 (float) 及其运算
 - 数学运算 : +, -, *, /
 - 赋值运算 : =, +=, -=, *=, /=

- 布尔型 (bool)
 - 只有两个值True和False
- 运算
 - 与 (and)
 - 或 (or)
 - 非 (not)

真值表：

not	True	False
	False	True

and	True	False
True	True	False
False	False	False

or	True	False
True	True	True
False	True	False

变量及运算

- 变量名 – 代码风格
 - 程序员必修课
 - 平时多注意，多积累
 - 参考：<https://google.github.io/styleguide/pyguide.html>

a = 1

number_of_days = 7

程序控制流 Control flow

- 顺序结构
 - 从上到下顺序执行语句
- 控制结构
- 循环结构

控制结构 - if语句

- 语句一

```
if condition:  
    // condition is true
```

- 语句二

```
if condition:  
    // condition is true  
else:  
    // condition is false
```

控制结构 - if语句

- 语句三

```
if condition1:  
    // condition1 is true  
elif condition2:  
    // condition1 is false and condition2 is true  
else:  
    // condition1 is false and condition2 is false
```

- 缩进

- 连续相同缩进的代码在同一个代码块中
- 缩进的空白数量是**可变的**，本课要求使用**4空格**

控制结构 - if语句

- 练习一
 - Max of 3 Numbers
 - <http://www.lintcode.com/en/problem/max-of-3-numbers/>
 - <https://www.jiuzhang.com/solution/max-of-3-numbers/>

循环结构

- 先看一个数据结构：列表（list）
 - 用一个变量存储很多数据

```
scores = [90, 95, 92, 89, 100, 98]
```

```
my_score = scores[0]
```

```
scores[0] = 100
```

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)
```

循环结构

- break语句
 - 强制退出循环

```
1 target = 89
2 found = False
3
4 for score in scores:
5     if score == target:
6         found = True
7         break
```

循环结构

- **continue语句**
 - 提前结束本轮循环

```
1 for score in scores:  
2     if score < 60:  
3         continue  
4     print(score)
```

循环结构

- while 循环

```
1 i = 0                                循环变量
2 while i < len(scores):
3     print(scores[i])
4     i += 1
```

- while 循环与for 循环完全等价

循环结构

- 多重循环

```
1 multi_scores = [[95, 88, 54], [67, 25]]  
2  
3 for scores in multi_scores:  
4     for score in scores:  
5         print(score)  
6  
7 for i in range(len(multi_scores)):  
8     for j in range(len(multi_scores[i])):  
9         print(multi_scores[i][j])
```

函数 Function

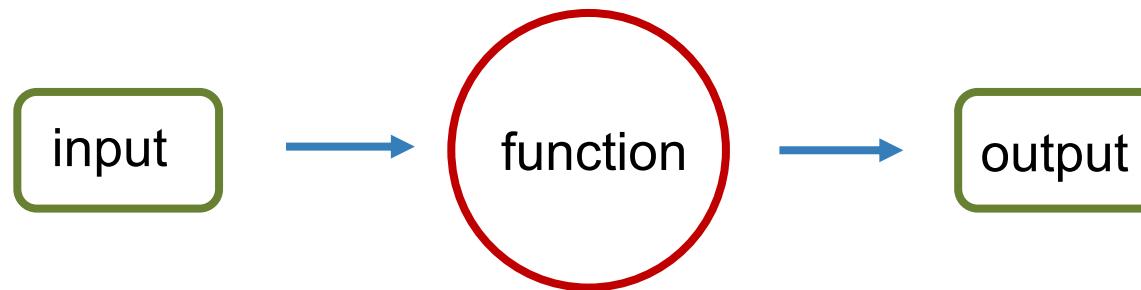
函数 Function

- 函数：具有特定功能的代码段
- Why?
 - 增加代码复用 (code reuse)
 - 增强程序可读性 (readability)

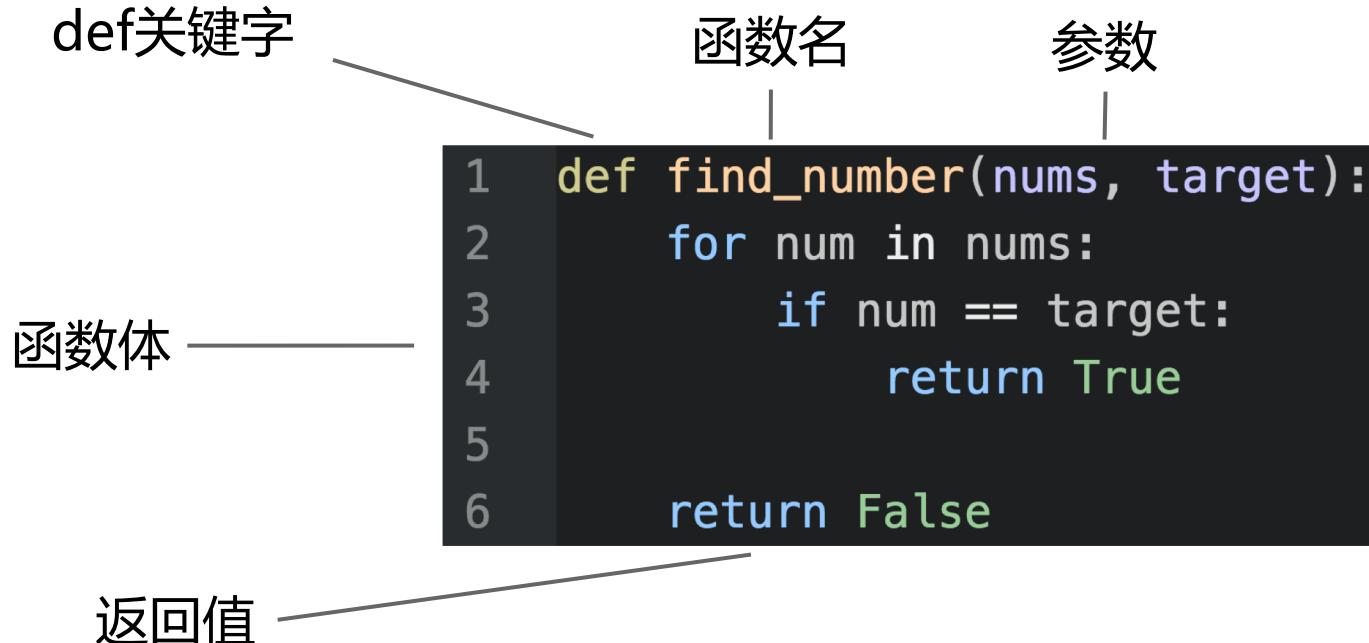
Python有很多内建的函数，如 `print()`，用户也可以自己定义函数，称之为用户自定义函数。

函数 Function

- 函数
 - 广义来讲，有输入有输出就是函数



函数 Function



- 函数的调用
 - 给定输入，返回输出
 - 程序执行流程（演示）
 - 把函数的输出当做一个值来使用
- 函数的参数传递

- 练习二
 - Swap two integers in an array using function
 - <http://www.lintcode.com/en/problem/swap-two-integers-in-array/>
 - <https://www.jiuzhang.com/solution/swap-two-integers-in-array/>

面向对象入门

— 类和对象

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

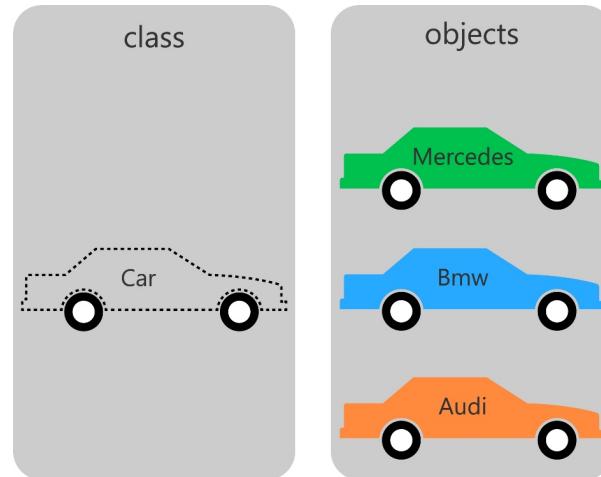
- 什么是对象 (Object) ?
 - 在面向对象的世界观中 : 世间万物皆为对象
 - 属性 & 行为
 - 小狗
 - 汽车
 - 电脑

类和对象

- 对象的属性
 - 小狗：四条腿
 - 汽车：四个轮子，方向盘
 - 电脑：屏幕，键盘
- 对象的行为
 - 小狗：汪汪叫
 - 汽车：加减速，转弯
 - 电脑：运行程序，播放视频

类和对象

- 什么是类 (class) ?
 - 类是对象的蓝图



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

- `__init__()` 是构造函数，当创建这个类的对象时自动调用
- `self` 指的是对象本身，`self` 在定义类的方法时是必须有的，但是调用时看不到这个参数

```
1  class Student():
2
3      def __init__(self, name, score):
4          self.name = name
5          self.score = score
```

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

类和对象

- 成员变量 (member variable)
 - 又叫域 (field)
 - 表示对象的属性
 - 命名规则和变量名相同

类和对象

- 成员函数 (member function)
 - 又叫方法 (method)
 - 表示对象的行为
 - 命名规则和函数名相同

类和对象

- 什么是OOP ?
 - 用对象构建程序
- 面向对象的三大特征
 - 封装 (Encapsulation)
 - 继承 (Inheritance)
 - 多态 (Polymorphism)

面向对象

- 封装性
 - 将属性和行为**封装**成一个类，并尽可能**隐蔽**类（对象）的内部细节，对外形成一个边界，只保留有限的**对外接口**使之与外部发生联系
- 代码演示



- 封装性
 - 改变程序的组织方式
 - 增加代码的复用率
 - 提高程序开发效率

面向对象

- 构造函数 (constructor)
- field和method的访问权限
- setter和getter

面向对象

- 练习三
 - Student id
 - <http://www.lintcode.com/en/problem/student-id/>
 - <https://www.jiuzhang.com/solution/student-id/>

总结

- 程序控制流
 - 顺序结构
 - 控制结构
 - 循环结构
- 函数的定义和调用
- 面向对象入门



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



谢谢大家