

# 第一章 Python语法基础 I

林平之 老师



对课程有疑问？购买时遇到问题？获取更多优惠信息？  
扫一扫，在线咨询

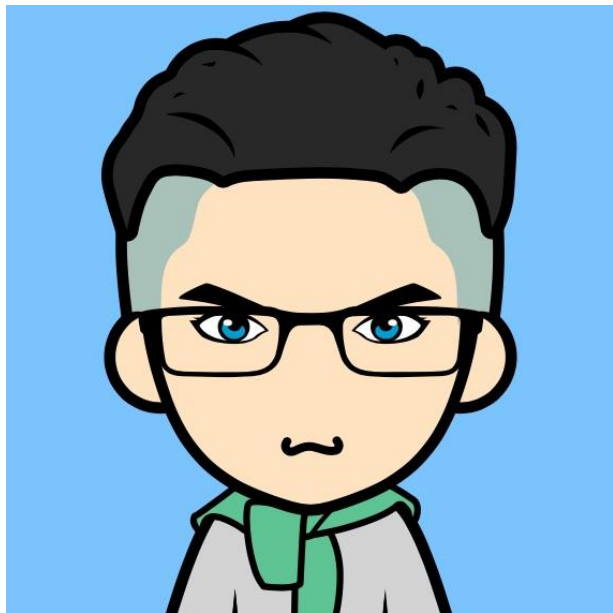
微信: [ninechapter](#)

知乎专栏: <http://zhuankan.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



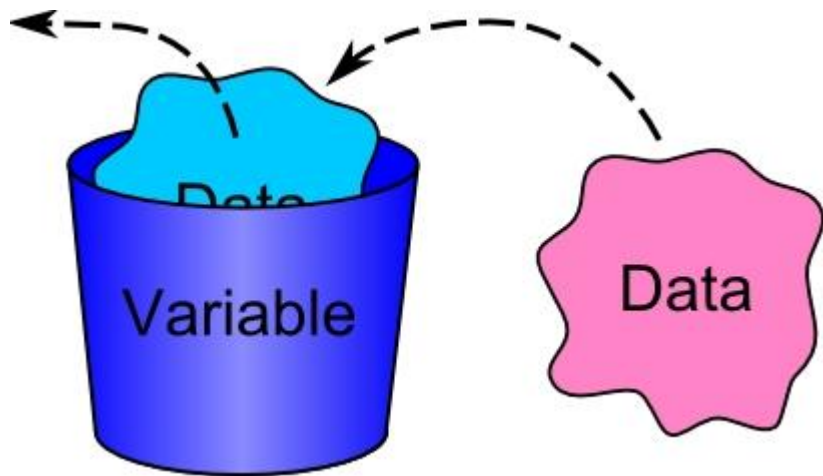
林老师

全国算法竞赛一等奖  
国内TOP2名校毕业  
参加国家信息学竞赛NOI  
前FLAG工程师  
拥有丰富的面试经验

- Python的基础类型
- 程序控制流
- 函数的定义及调用
- Python内置数据结构一

# Python的基础类型

- 变量(variable): 存储数据的容器
  - 赋值运算相当于把数据装入变量中



- Python变量无类型，但里面的数据有类型
- 动态类型语言 (Dynamically typed language) vs.  
静态类型语言 (Statically typed language)

- 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>

变量名: 英文字母、数字和下划线的组合, 且不能用数字开头

~~a = 1~~

number\_of\_days = 7



- Python中的整数
  - -10, 1, 100
  - `2 ** 1000` # Python的整数可以非常大
  - `0x1f0` # 十六进制以0x前缀和0-9, a-f表示
- 运算
  - 数学运算: +, -, \*, /, //, %

- 交换两个变量的值

```
# 交换两个变量的值
```

```
num1, num2 = 10, 15
```

```
# NP
```

```
tmp = num1
```

```
num1 = num2
```

```
num2 = tmp
```

```
# P
```

```
num1, num2 = num2, num1
```

- Python中的浮点数(小数)
  - -1.1, -2.0
  - 1.2e10, 1.2e-5 # 科学计数法
- 运算
  - 数学运算: +, -, \*, /

- Python中的布尔型
  - 只有两个值True和False
- 表达式

```
num1, num2, num3 = 10, 15, 20  
num1 == 10  
num2 != 13  
num1 <= num2 < num3
```

- 运算

- 与 ( and )
- 或 ( or )
- 非 ( not )

- 运算优先级

- 非 > 与 > 或

真值表：

not    True   False

False   True

and    True   False

or    True   False

True   True   False

True   True   True

False   False   False

False   True   False

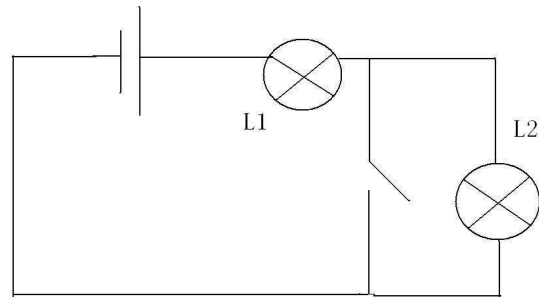
- 短路(Short-circuit)运算

```
num = 5
```

```
float_num = 3.1
```

```
or_true = num == 5 or float_num > 4.5
```

```
and_false = num > 8 and float_num < 4.2
```



# 程序控制流 Control Flow

- 顺序结构
  - 从上到下顺序执行语句
- 控制/分支结构
- 循环结构



- 语句一

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

- 语句二

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

- 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空格或tab**
- 三元运算符(Ternary operator)

```
# 三元运算符
```

```
num1 = 13
```

```
# NP
```

```
if num1 > 10:
```

```
    num2 = 20
```

```
else:
```

```
    num2 = 5
```

```
# P
```

```
num2 = 20 if num1 > 10 else 5
```

- 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
```

- 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])
```

- 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)
```

- 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
```

- 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)
```



- while 循环

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

循环变量

- while 循环与for 循环完全等价

- 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: 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。

def关键字

函数名

参数

函数体

```
1 def find_number(nums, target):  
2     for num in nums:  
3         if num == target:  
4             return True  
5  
6     return False
```

返回值

- 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

- 赠送基础课程



九章算法-Justin

United States



Scan the QR code to add me on WeChat

- 项目介绍

— [https://mp.weixin.qq.com/s/rgbhwmVJuYCU2IWPDi6\\_dg](https://mp.weixin.qq.com/s/rgbhwmVJuYCU2IWPDi6_dg)

# Python内置数据结构一



- 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



- 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']
```

- 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

- 列表生成式(List Comprehensions)
  - 生成100以内5的倍数形成的list

*# 生成100以内5的倍数的数形成一个List*

*# NP*

```
result = []  
for i in range(101):  
    if i % 5 == 0:  
        result.append(i)
```

*# P*

```
result = [i for i in range(101) if i % 5 == 0]
```

- 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,)
```

- 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?

- 互相转化
  - `tuple([1, 2, 3])`  $\square$  `(1, 2, 3)`
  - `list((1, 2, 3))`  $\square$  `[1, 2, 3]`

- 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()`

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

- 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()
```

- 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)

- 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

- 翻转字符串

- <https://www.lintcode.com/problem/reverse-string/description>

```
# 翻转字符串
```

```
s = 'jiuzhang'
```

```
# NP
```

```
result = ''
```

```
for i in range(len(s)-1, -1, -1):
```

```
    result += s[i]
```

```
# P
```

```
result = s[::-1]
```

- 拼接列表中的字符串

*# 拼接列表中的字符串*

```
strs = ['Zhang', 'is', 'nice']
```

*# NP*

```
result = ''
```

```
for s in strs:
```

```
    result += s + ' '
```

*# P*

```
result = ' '.join(strs)
```

- Python的基础类型
- 程序控制流
- 函数的定义及调用
- Python内置数据结构一





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)



## 课程顾问微信

获取更多优惠和相关礼包  
请咨询九章课程顾问老师

<http://jiuzhangai.mikecrm.com/rUPWXua>

谢谢大家