Python for Senior

Lesson 1 v1.0.0

2016.8 by David.Yi

复习

Python 初级中的一些基本概念

- input 和 print 用法
- 循环语句 for
- 数据类型 list
- 数据类型 dict
- 函数的写法和用法

input 和 print

- 使用 print()，在括号中加上字符串，就可以在屏幕上输出指定的文字
- 使用 input()，可以让用户输入字符串，并存放到一个变量里

想一想这个执行结果是什么？

```
name = input('What is your name? ')
print('Hello ' + name)
```

循环语句 for

- for x in ... 循环就是把每个元素代入变量x，然后执行缩进块的语句
- 缩进块，是 python 中表示语句层次关系的重要形式
- range() 是我们常用的产生一个整数序列的函数，在 for 循环中广泛使用
- 变量的命名、计算、类型

想一想这个执行结果是什么？

```
j = 0
for i in range(10,20,2):
    j = j + i
print(i+j)
```

数据类型 list

- list 是一种有序的集合，可以随时添加和删除其中的元素，是 python 中使用率非常高的数据类型。
- list 的定义
- 访问 list 中的元素
- list 追加元素、插入元素
- list 删除元素
- list 长度计算
- list 切片
- 数据类型 tuple 的相似性，元素定义后不可改变

数据类型 list 举例

想一想这个执行结果是什么？

```
a = [1,2,5,9]
a.append(4)
a.pop(2)
a.remove(1)

j = 0
for i in range(len(a)):
    j = j + a[i]

print(j)
```

数据类型 dict

- dict 使用键-值（key-value）存储内容，具有极快的查找速度，是 python 开发中常用的数据类型
- dict 的定义，理解 key 和 value
- 访问 dict 中的 value
- 在 dict 中判断 key 是否存在
- 在 dict 中删除 key
- 数据类型 set 的相似性，可以看成数学意义上的无序和无重复元素的集合

函数

- 函数是组织好的，可重复使用的，用来实现单一，或相关联功能的代码段。函数能提高应用的模块性，和代码的重复利用率。
- 函数的代码结构：函数名，参数，返回值
- 要学会用和写函数，真正的编程是离不开函数的，函数也是面向对象编程方法的基础

函数举例

想一想这个执行结果是什么？

```
def my_max_min(*number):
    a_list = list(number)
    a_list.sort()
    my_max = a_list[-1]
    my_min = a_list[0]
    print(a_list)
    return my_max, my_min

print(my_max_min(1,5,30,14,25))
```

Python 中级课程大纲

我们会学习以下主要内容

- 基本语句进阶
- 函数进阶
- 列表进阶，列表生成器
- 类的入门
- 更多 python 标准库
- 更多实际功能程序开发和讨论
- 编程思路的培养和提高

Python 中级和初级的区别

- 进一步理解现代编程语言的巧妙之处
- 阅读和编制更长和更加复杂的程序
- 对逻辑思维更好的锻炼和挑战
- 或许会有同学跟不上进度，没有关系，对于知识的理解有时候是顿悟和跳跃式的
- 建议部分同学可以养成继续学习和自学编程的能力，可能的话和学校项目有所结合，真正解决一些问题

if 语句进阶

- Python 没有类似其他语言 switch 或者 case 的语句，python 保持一贯的简洁态度来看待多种条件判断的情况
- if...elif...else, 可以用多个并列的 elif 来判断多种条件

```
n = int(input('Please input an integer:'))

if n < 0 :
    print('Negative')
elif n >= 0 and n < 100:
    print('Between 0..100')
elif n >= 100 and n < 200:
    print('Between 100..200')
else :
    print('More')
```

if 语句进阶

- 思考一下：如果不用 elif 是否可以完成刚才的要求？
- 结论：使用 if…elif…else 可以让程序结构更加清楚，减少缩进，提高程序可读性

- 进一步：三元操作符 c?x:y，c 为表达式，x 是 c 为 True 时的结果，y 是 c 为 False 时的结果

举例

- list 排序按照元素的长度，而不是默认按照字母顺序。比如 [‘Beijing’, ‘Tokyo’, ‘Shanghai’] 排序后是 [‘Beijing’, ‘Shanghai’, ‘Tokyo’]

- 找到符合这样规律的整数 x*x + y*y = z*z, 比如 3*3+4*4=5*5，找到设定某个范围内所有这样的整数