李隆康 Python 学习笔记：

（布尔类型：只有 True False,注意大小写）

1 月 15 号：

1. 注释：#表示注释
2. 输出：print ***；
3. 缩进：Python 的代码块不使用大括号（{}）来控制类，函数以及其他逻辑判断。python 最具特色的就是用缩进来写模块。缩进的空白数量是可变的，但是所有代码块语句必须包含相同的缩进空白数量，这个必须严格执行。
4. 多行：Python 可以使用斜杠（ \）将一行的语句分为多行显示，如下所示：

```
total = item_one + \
        item_two + \
        item_three;
```

语句中包含[], {} 或 () 括号就不需要使用多行连接符。如下实例：

```
days = ['Monday', 'Tuesday', 'Wednesday',
        'Thursday', 'Friday'];
```

5. 引号：单引号（‘’）表示一个单词；双引号（“”）表示一句话；三引号（“‘’”）表示多行；例如：

```
word = 'word'
sentence = "这是一个句子。"
paragraph = """这是一个段落。
包含了多个语句"""
```

6. 输入：通过 raw_input() 来输入

```
year = int(raw_input('year:\n'))
month = int(raw_input('month:\n'))
day = int(raw_input('day:\n'))
```

1 月 17 号：

1. 变量：Python 中的变量赋值不需要类型声明；（类似 JavaScript）

   每个变量在内存中创建；

   每个变量在使用前都必须赋值，只有赋值以后才会被创建。
2. 赋值：以下赋值都可以

```
counter = 100 # 赋值整型变量
miles = 1000.0 # 浮点型
name = "John" # 字符串
```

```
a = b = c = 1
```

```
a, b, c = 1, 2, "john"
```

3. 数据：

   标准数据类型：5 种（数字、字符串、列表、元组、字典）

   数字：int long float complex（注意：没有 double， 多了个复数）

   字符串：（用引号 ‘’ “” “‘’”）

```
str = 'Hello World!'

print str           # 输出完整字符串
print str[0]        # 输出字符串中的第一个字符
print str[2:5]      # 输出字符串中第三个至第五个之间的字符串
print str[2:]       # 输出从第三个字符开始的字符串
print str * 2       # 输出字符串两次
print str + "TEST"  # 输出连接的字符串
```

```
Hello World!
H
llo
llo World!
Hello World!Hello World!
Hello World!TEST
```

（尤其是输出两次的方式!!!）

列表：（用中括号[]）

```
list = [ 'runoob', 786 , 2.23, 'john', 70.2 ]
tinylist = [123, 'john']

print list                # 输出完整列表
print list[0]             # 输出列表的第一个元素
print list[1:3]           # 输出第二个至第三个的元素
print list[2:]            # 输出从第三个开始至列表末尾的所有元素
print tinylist * 2        # 输出列表两次
print list + tinylist     # 打印组合的列表
```

```
['runoob', 786, 2.23, 'john', 70.2]
runoob
[786, 2.23]
[2.23, 'john', 70.2]
[123, 'john', 123, 'john']
['runoob', 786, 2.23, 'john', 70.2, 123, 'john']
```

元组：（用小括号（））相当于只读列表，不能二次赋值

```
tuple = ( 'runoob', 786 , 2.23, 'john', 70.2 )
tinytuple = (123, 'john')

print tuple               # 输出完整元组
print tuple[0]            # 输出元组的第一个元素
print tuple[1:3]          # 输出第二个至第三个的元素
print tuple[2:]           # 输出从第三个开始至列表末尾的所有元素
print tinytuple * 2       # 输出元组两次
print tuple + tinytuple   # 打印组合的元组
```

```
('runoob', 786, 2.23, 'john', 70.2)
runoob
(786, 2.23)
(2.23, 'john', 70.2)
(123, 'john', 123, 'john')
('runoob', 786, 2.23, 'john', 70.2, 123, 'john')
```

字典：（用大括号{}）完整的字典用大括号，部分内容用中括号；

列表是有序的对象结合，字典是无序的对象集合；

字典通过键值对的方式来存储；

```
dict = {}
dict['one'] = "This is one"
dict[2] = "This is two"

tinydict = {'name': 'john','code':6734, 'dept': 'sales'}


print dict['one']           # 输出键为'one' 的值
print dict[2]               # 输出键为 2 的值
print tinydict              # 输出完整的字典
print tinydict.keys()       # 输出所有键
print tinydict.values()     # 输出所有值
```

```
This is one
This is two
{'dept': 'sales', 'code': 6734, 'name': 'john'}
['dept', 'code', 'name']
['sales', 6734, 'john']
```

1 月 19 号：（循环语句 条件语句）!!! 注意：Python 里面每句话后面是没有分号；的!!!

1. if – else 语句：（尤其注意格式!!! 在 if else 后面有冒号：）

```
flag = False
name = 'luren'
if name == 'python':          # 判断变量否为'python'
    flag = True            # 条件成立时设置标志为真
    print 'welcome boss'     # 并输出欢迎信息
else:
    print name               # 条件不成立时输出变量名称
```

```
>>> luren                          # 输出结果
```

多条件输出：（没有 switch，没有 else if ，只有 elif）

```
num = 5
if num == 3:             # 判断num的值
    print 'boss'
elif num == 2:
    print 'user'
elif num == 1:
    print 'worker'
elif num < 0:            # 值小于零时输出
    print 'error'
else:
    print 'roadman'      # 条件均不成立时输出
```

2. 或与：（不用数学符号&& ||，用文字 and or）

```
num = 9
if num >= 0 and num <= 10:    # 判断值是否在0~10之间
    print 'hello'
>>> hello                # 输出结果

num = 10
if num < 0 or num > 10:    # 判断值是否在小于0或大于10
    print 'hello'
else:
        print 'undefine'
>>> undefine             # 输出结果
```

3. while 循环：

```
count = 0
while (count < 9):
   print 'The count is:', count
   count = count + 1

print "Good bye!"
```

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
The count is: 6
The count is: 7
The count is: 8
Good bye!
```

4. while – else：（只有 Python 可以这样）

```
count = 0
while count < 5:
   print count, " is  less than 5"
   count = count + 1
else:
   print count, " is not less than 5"
```

```
0 is less than 5
1 is less than 5
2 is less than 5
3 is less than 5
4 is less than 5
5 is not less than 5
```

5. for 循环：（类似 Java， 注意 in 的用法）

```
for letter in 'Python':     # 第一个实例
   print '当前字母 :', letter

fruits = ['banana', 'apple',  'mango']
for fruit in fruits:        # 第二个实例
   print '当前字母 :', fruit

print "Good bye!"
```

```
当前字母 : P
当前字母 : y
当前字母 : t
当前字母 : h
当前字母 : o
当前字母 : n
当前字母 : banana
当前字母 : apple
当前字母 : mango
Good bye!
```

内置函数：len()　　#返回列表的长度

Range()　#返回一个序列的数

（以下实例通过序列索引迭代）

```
fruits = ['banana', 'apple',  'mango']
for index in range(len(fruits)):
   print '当前水果 :', fruits[index]

print "Good bye!"
```

```
当前水果 : banana
当前水果 : apple
当前水果 : mango
Good bye!
```

6. for-else 循环：（只有 Python 这种搭配）注意：以下的 else 不是和 if 并列，而是和 for

```
for num in range(10,20):  # 迭代 10 到 20 之间的数字
   for i in range(2,num): # 根据因子迭代
      if num%i == 0:      # 确定第一个因子
         j=num/i          # 计算第二个因子
         print '%d 等于 %d * %d' % (num,i,j)
         break            # 跳出当前循环
   else:                  # 循环的 else 部分
      print num, '是一个质数'
```

```
10 等于 2 * 5
11 是一个质数
12 等于 2 * 6
13 是一个质数
14 等于 2 * 7
15 等于 3 * 5
16 等于 2 * 8
17 是一个质数
18 等于 2 * 9
19 是一个质数
```

1 月 24 号：（函数 模块）

1. 函数： 内建函数、自定义函数；

规则：函数代码块以 def 关键词开头，后接函数标识符名称和圆括号()。
函数的第一行语句可以选择性地使用文档字符串—用于存放函数说明。
return [表达式] 选择性地返回一个值；不带表达式的 return 相当于返回 None。

举例 1：

```
# 定义函数
def printme( str ):
   "打印任何传入的字符串"
   print str;
   return;

# 调用函数
printme("我要调用用户自定义函数!");
printme("再次调用同一函数");
```

```
我要调用用户自定义函数!
再次调用同一函数
```

举例 2：（Python 所有的参数都是按引用传递，所以一旦函数内改变，函数外都会变）

```
# 可写函数说明
def changeme( mylist ):
   "修改传入的列表"
   mylist.append([1,2,3,4]);
   print "函数内取值: ", mylist
   return

# 调用changeme函数
mylist = [10,20,30];
changeme( mylist );
print "函数外取值: ", mylist
```

```
函数内取值:  [10, 20, 30, [1, 2, 3, 4]]
函数外取值:  [10, 20, 30, [1, 2, 3, 4]]
```

举例 3：（关键字参数顺序在声明和调用的时候可以不一致）

```
#可写函数说明
def printinfo( name, age ):
   "打印任何传入的字符串"
   print "Name: ", name;
   print "Age ", age;
   return;

#调用printinfo函数
printinfo( age=50, name="miki" );
```

```
Name:  miki
Age  50
```

举例 4：（缺省参数，在声明时初始化，当调用时没有赋值也没关系）

```
#可写函数说明
def printinfo( name, age = 35 ):
   "打印任何传入的字符串"
   print "Name: ", name;
   print "Age ", age;
   return;

#调用printinfo函数
printinfo( age=50, name="miki" );
printinfo( name="miki" );
```

```
Name:  miki
Age  50
Name:  miki
Age  35
```

举例 5：（匿名函数 lambda）

```
# 可写函数说明
sum = lambda arg1, arg2: arg1 + arg2;

# 调用sum函数
print "相加后的值为 : ", sum( 10, 20 )
print "相加后的值为 : ", sum( 20, 20 )
```

```
相加后的值为 :  30
相加后的值为 :  40
```

举例 6:（局部变量和全局变量，虽然同名，但是一个全局、一个局部，在这视为不同。）

```
total = 0; # 这是一个全局变量
# 可写函数说明
def sum( arg1, arg2 ):
   #返回2个参数的和."
   total = arg1 + arg2; # total在这里是局部变量.
   print "函数内是局部变量 : ", total
   return total;

#调用sum函数
sum( 10, 20 );
print "函数外是全局变量 : ", total
```

```
函数内是局部变量 :  30
函数外是全局变量 :  0
```