# Python简单介绍

## 1.1 简单发展史

1989年的圣诞节，Guido开始编写Python语言的编译器。Python这个名字，来自Guido所挚爱的电视剧Monty Python’s Flying Circus。他希望这个新的叫做Python的语言，能符合他的理想：创造一 种C和shell之间，功能全面，易学易用，可拓展的语言。

1991年，第一个Python编译器诞生。它是用C语言实现的，并能够调用C语言的库文件。从一出生，Python已经具有了：类，函数，异常处理，包含表和词典在内的核心数据类型，以及模块为基础的拓 展系统。

2001年，发布了python 2.x版本，版本更新至2.7，2.x内不再更新2.7以上版本；

2013年，发布了python 3.0版本；

## 1.2 优缺点

### 1.2.1优点

1）“优雅”、“明确”、“简单”

2）开发效率 高

3）高级语言

4）可移植性

5）可扩展性

6）可嵌入型

## 1.2.2 缺点

1）代码不可加密

2）多线程支持一般

3）慢

## 1.3 适合的领域

1）web网站和各种网络服务

2）系统工具和脚本

3）包装其他语言的胶水语言

# Python 入门级语法

## 2.1变量和数据类型

## Python中数据类型

**一、整数**

Python可以处理任意大小的整数，当然包括负整数，在Python程序中，整数的表示方法和数学上的写法一模一样，例如：1，100，-8080，0，等等。

计算机由于使用二进制，所以，有时候用十六进制表示整数比较方便，十六进制用0x前缀和0-9，a-f表示，例如：0xff00，0xa5b4c3d2，等等。

**二、浮点数**

浮点数也就是小数，之所以称为浮点数，是因为按照科学记数法表示时，一个浮点数的小数点位置是可变的，比如，1.23x10^9和12.3x10^8是相等的。浮点数可以用数学写法，如1.23，3.14，-9.01，等等。但是对于很大或很小的浮点数，就必须用科学计数法表示，把10用e替代，1.23x10^9就是**1.23e9**，或者**12.3e8**，0.000012可以写成**1.2e-5**，等等。

**三、字符串**

字符串是以''或""括起来的任意文本，比如**'abc'**，**"xyz"**等等。请注意，**''**或**""**本身只是一种表示方式，不是字符串的一部分，因此，字符串'abc'只有a，b，c这3个字符。

**四、布尔值**

布尔值和布尔代数的表示完全一致，一个布尔值只有True、False两种值，要么是True，要么是False，在Python中，可以直接用True、False表示布尔值（请注意大小写），也可以通过布尔运算计算出来。

布尔值可以用and、or和not运算。

**五、空值**

空值是Python里一个特殊的值，用None表示。None不能理解为0，因为0是有意义的，而None是一个特殊的空值。

## Python之print语句

**print**语句可以向屏幕上输出指定的文字。比如输出'hello, world'，用代码实现如下：

>>> print 'hello, world'

**注意：**

1.当我们在Python交互式环境下编写代码时，>>>是Python解释器的提示符，不是代码的一部分。

2.当我们在文本编辑器中编写代码时，千万不要自己添加 >>>。

print语句也可以跟上多个字符串，用逗号“,”隔开，print也可以打印整数，或者计算结果：

>>> print 300

300 #运行结果

>>> print 100 + 200

300 #运行结果

## Python的注释

任何时候，我们都可以给程序加上注释。注释是用来说明代码的，给自己或别人看，而程序运行的时候，Python解释器会直接忽略掉注释，所以，有没有注释不影响程序的执行结果，但是影响到别人能不能看懂你的代码。

Python的注释以 # 开头，后面的文字直到行尾都算注释

## Python中什么是变量

在Python程序中，变量是用一个变量名表示，变量名必须是大小写英文、数字和下划线（\_）的组合，且不能用数字开头，比如：

a = 1

变量a是一个整数。

在Python中，等号=是赋值语句，可以把任意数据类型赋值给变量，同一个变量可以反复赋值，而且可以是不同类型的变量

## Python中定义字符串

前面我们讲解了什么是字符串。字符串可以用''或者""括起来表示。

对字符串的某些特殊字符进行“转义”，Python字符串用\进行转义。

## Python中raw字符串与多行字符串

如果一个字符串包含很多需要转义的字符，对每一个字符都进行转义会很麻烦。为了避免这种情况，我们可以在字符串前面加个前缀 r ，表示这是一个 raw 字符串，里面的字符就不需要转义了。例如：

r'\(~\_~)/ \(~\_~)/'

如果要表示多行字符串，可以用'''...'''表示：

'''Line 1

Line 2

Line 3'''

## Python中Unicode字符串

如果中文字符串在Python环境下遇到 UnicodeDecodeError，这是因为.py文件保存的格式有问题。可以在第一行添加注释

# -\*- coding: utf-8 -\*-

## Python中整数和浮点数

Python支持对整数和浮点数直接进行四则混合运算，运算规则和数学上的四则运算规则完全一致。

## Python中布尔类型

**涉及到 and 和 or 运算的一条重要法则：短路计算。**

Python把0、空字符串''和None看成 False，其他数值和非空字符串都看成 True，

## 2.2 List和Tuple

## Python创建list

Python内置的一种数据类型是列表：list。list是一种有序的集合，可以随时添加和删除其中的元素。

构造list非常简单，按照上面的代码，直接用 [ ] 把list的所有元素都括起来，就是一个list对象。通常，我们会把list赋值给一个变量，这样，就可以通过变量来引用list：

由于Python是动态语言，所以list中包含的元素并不要求都必须是同一种数据类型，我们完全可以在list中包含各种数据：

## Python按照索引访问list

**需要特别注意的是**，索引从 0 开始，也就是说，第一个元素的索引是0，第二个元素的索引是1，以此类推。

使用索引时，**千万注意不要越界**。

## Python之倒序访问list

倒数第一 -1，倒数第二用 -2 表示，倒数第三用 -3 表示，倒数第四用 -4 等等：

## Python之添加新元素

第一个办法是用 list 的 append() 方法，追加到 list 的末尾。

方法是用list的 insert(int，object)方法，它接受两个参数，第一个参数是索引号，第二个参数是待添加的新元素，加到索引号前面

## Python从list删除元素

**pop()**方法总是删掉list的最后一个元素，并且它还返回这个元素

**pop(int** )删除该索引的数值并返回

，

## Python中替换元素

直接赋值替换

## ython之创建tuple

tuple是另一种有序的列表，中文翻译为“ 元组 ”。tuple 和 list 非常类似，但是，tuple一旦创建完毕，就不能修改了。

包含 0 个元素的 tuple，也就是空tuple，直接用 ()表示：

>>> t = ()

>>> print t

()

创建包含1个元素的 tuple 呢？来试试：

>>> t = (1)

>>> print t

1

好像哪里不对！t 不是 tuple ，而是整数1。**为什么**呢？

因为()既可以表示tuple，又可以作为括号表示运算时的优先级，结果 (1) 被Python解释器计算出结果 1，导致我们得到的不是tuple，而是整数 1。

正是因为用()定义单元素的tuple有歧义，所以 Python 规定，单元素 tuple 要多加一个逗号“,”，这样就避免了歧义：

>>> t = (1,)

>>> print t

(1,)

Python在打印单元素tuple时，也自动添加了一个“,”，为了更明确地告诉你这是一个tuple。

## Python之“可变”的tuple

前面我们看到了tuple一旦创建就不能修改。现在，我们来看一个“可变”的tuple：

>>> t = ('a', 'b', ['A', 'B'])

包含的一个list，tuple的元素没有改变，但是list的元素可以改变的。

可以改成>>> t = ('a', 'b',('A', 'B'))

## 2.2 条件判断和循环

## Python之if语句

**:**Python代码的缩进规则。具有相同缩进的代码被视为代码块，上面的3，4行 print 语句就构成一个代码块（但不包括第5行的print）。如果 if 语句判断为 True，就会执行这个代码块。

缩进请严格按照Python的习惯写法：4个空格，不要使用Tab，更不要混合Tab和空格，否则很容易造成因为缩进引起的语法错误。

如果你在Python交互环境下敲代码，还要特别留意缩进，并且退出缩进需要多敲一行回车：

## ython之 if-else

当 if 语句判断表达式的结果为 True 时，就会执行 if 包含的代码块

利用 if ... else ... 语句，我们可以根据条件表达式的值为 True 或者 False ，分别执行 if 代码块或者 else 代码块。

**注意:** else 后面有个“:”。

## Python之 if-elif-else

很多时候一个if…else不够用的，可以使用 if ... 多个elif ... else ... 的结构，一次写完所有的规则

elif 意思就是 else if。这样一来，我们就写出了结构非常清晰的一系列条件判断。

## Python之 for循环

Python的 for 循环就可以依次把list或tuple的每个元素迭代出来：

L = ['Adam', 'Lisa', 'Bart']

for name in L:

print name

**注意:** name 这个变量是在 for 循环中定义的，意思是，依次取出list中的每一个元素，并把元素赋值给 name，然后执行for循环体（就是缩进的代码块）。

## Python之 while循环

和 for 循环不同的另一种循环是 while 循环，while 循环不会迭代 list 或 tuple 的元素，而是根据表达式判断循环是否结束。

## Python之 break退出循环

用 for 循环或者 while 循环时，如果要在循环体内直接退出循环，可以使用 break 语句。

## Python之 continue继续循环

在循环过程中，可以用break退出当前循环，还可以用continue跳过后续循环代码，继续下一次循环。

## Python之 多重循环

在循环内部，还可以嵌套循环