

# Python语言十分钟快速入门

2015-11-03 数聚

小提示

点击上方蓝色字体即可一键关注哦~

---

## 如何关注

数聚 微信号：shuju\_vip

1、复制“微信号或ID”，在“添加朋友”中粘贴搜索号码关注。

2、点击微信右上角的“+”，会出现“添加朋友”，进入“查找公众号”，输入公众号名“数聚”，即可找到。

【数聚】是一款大数据知识分享与社交平台！我们的编辑团队均来自互联网行业，核心成员均有10年以上数据从业经验，我们保证：用心创作或推荐每篇文章！欢迎大家私信留言告诉我们您最想了解的主题！我们期望能够和您一起打造最懂大数据行业的交流平台！

---

【作者】 Rocrocket Wu

【来源】 <http://roclinux.cn/?p=2338>

### 【简介】

Python（蟒蛇）是一种动态解释型的编程语言。

Python可以在Windows、UNIX、MAC等多种操作系统上使用，也可以在Java、.NET开发平台上使用。



python logo

### 【特点】

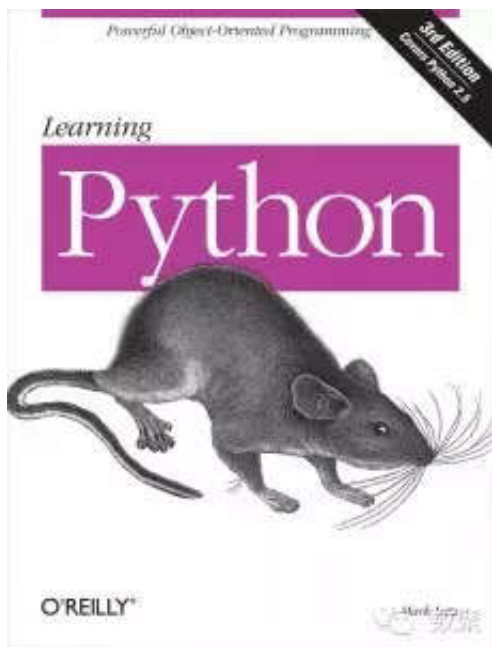
1 Python使用C语言开发，但是Python不再有C语言中的指针等复杂的数据类型。

2 Python具有很强的面向对象特性，而且简化了面向对象的实现。它消除了保护类型、抽象类、接口等面向对象的元素。

3 Python代码块使用空格或制表符缩进的方式分隔代码。

4 Python仅有31个保留字，而且没有分号、begin、end等标记。

5 Python是强类型语言，变量创建后会对应一种数据类型，出现在统一表达式中的不同类型的变量需要做类型转换。



python book

### 【搭建开发环境】

1 可以到[www.python.org](http://www.python.org)下载安装包，然后通过 `configure`、`make`、`make install`进行安装。

2 也可以到[www.activestate.com](http://www.activestate.com)去下载ActivePython组件包。（ActivePython是对Python核心和常用模块的二进制包装，它是ActiveState公司发布的Python开发环境。ActivePython使得Python的安装更加容易，并且可以应用在各种操作系统上。ActivePython包含了一些常用的Python扩展，以及Windows环境的编程接口）。对ActivePython来说，如果你是windows用户，下载msi包安装即可；如果你是Unix用户，下载tar.gz包直接解压即可。

3 Python的IDE，包括PythonWin、Eclipse+PyDev插件、Komodo、EditPlus

### 【版本】

python2与python3是目前主要的两个版本。

如下两种情况下，建议使用python2：

- 1 你无法完全控制你即将部署的环境时；
- 2 你需要使用一些特定的第三方包或扩展时；

python3是官方推荐的且是未来全力支持的版本，目

前很多功能提升仅在python3版本上进行。

**【hello world】**

1 创建hello.py

2 编写程序:

```
1. if __name__ == '__main__':  
2.     print "hello word"
```

3 运行程序:

```
1. python ./hello.py
```

**【注释】**

1 无论是行注释还是段注释，均以#加一个空格来注释。

2 如果需要在代码中使用中文注释，必须在python文件的最前面加上如下注释说明：

```
1. # -*- coding: UTF-8 -*-
```

3 如下注释用于指定解释器

## 1. `#!/usr/bin/python`

### 【文件类型】

1 Python的文件类型分为3种，即源代码、字节代码和优化代码。这些都可以直接运行，不需要进行编译或连接。

2 源代码以.py为扩展名，由python来负责解释；

3 源文件经过编译后生成扩展名为.pyc的文件，即编译过的字节文件。这种文件不能使用文本编辑器修改。pyc文件是和平台无关的，可以在大部分操作系统上运行。如下语句可以用来产生pyc文件：

```
1. import py_compile
2. py_compile.compile( 'hello.py
    ' )
```

4 经过优化的源文件会以.pyo为后缀，即优化代码。它也不能直接用文本编辑器修改，如下命令可用来生成pyo文件：

```
1. python -O -m py_complie
```

## hello.py

### 【变量】

1 python中的变量不需要声明，变量的赋值操作即使变量声明和定义的过程。

2 python中一次新的赋值，将创建一个新的变量。即使变量的名称相同，变量的标识并不相同。用id()函数可以获取变量标识：

```
1. x = 1
2. print id(x)
3. x = 2
4. print id(x)
```

3 如果变量没有赋值，则python认为该变量不存在

4 在函数之外定义的变量都可以称为全局变量。全局变量可以被文件内部的任何函数和外部文件访问。

5 全局变量建议在文件的开头定义。

6 也可以把全局变量放到一个专门的文件中，然后通过import来引用：

gl.py文件中内容如下：

```
1.  _a = 1
2.  _b = 2
```

use\_global.py中引用全局变量：

```
1.  import gl
2.  def fun():
3.  print gl._a
4.  print gl._b
5.  fun()
```

## 【常量】

python中没有提供定义常量的保留字。可以自己定义一个常量类来实现常量的功能。

```
1.  class _const:
2.  class ConstError(TypeError):
    pass
3.  def
    __setattr__(self, name, vlaue):
4.  if
    self.__dict__.has_key(name):
5.  raise self.ConstError,
```



```
“Can’ t rebind
const(%s)” %name
6. self.__dict__[name]=value
7. import sys
8. sys.modules[__name__]=_const()
```

## 【数据类型】

1 python的数字类型分为整型、长整型、浮点型、布尔型、复数类型。

2 python没有字符类型

3 python内部没有普通类型，任何类型都是对象。

4 如果需要查看变量的类型，可以使用type类，该类可以返回变量的类型或创建一个新的类型。

5 python有3种表示字符串类型的方式，即单引号、双引号、三引号。单引号和双引号的作用是相同的。python程序员更喜欢用单引号，C/Java程序员则习惯使用双引号表示字符串。三引号中可以输入单引号、双引号或换行等字符。

## 【运算符和表达式】

1 python不支持自增运算符和自减运算符。例如  $i++/i-$  是错误的，但  $i+=1$  是可以的。

2  $1/2$ 在python2.5之前会等于0.5，在python2.5之后会等于0。

3 不等于为 $!=$ 或 $<>$

4 等于用 $==$ 表示

5 逻辑表达式中and表示逻辑与，or表示逻辑或，not表示逻辑非

## 【控制语句】

1 条件语句：

1. if (表达式) :
2. 语句1
3. else :
4. 语句2

2 条件语句：

1. if (表达式) :
2. 语句1
3. elif (表达式) :
4. 语句2
5. ...
6. elif (表达式) :
7. 语句n
8. else :
9. 语句m

### 3 条件嵌套:

1. if (表达式1) :
2. if (表达式2) :
3. 语句1
4. elif (表达式3) :
5. 语句2
6. ...
7. else:
8. 语句3
9. elif (表达式n) :
10. ...
11. else :
12. ...

### 4 python本身没有switch语句。

## 5 循环语句:

1. while(表达式) :
2. ...
3. else :
4. ...

## 6 循环语句:

1. for 变量 in 集合 :
2. ...
3. else :
4. ...

7 python不支持类似c的for(i=0;i<5;i++)这样的循环语句，但可以借助range模拟：

1. for x in range(0, 5, 2):
2. print x

## 【数组相关】

1 元组 (tuple) : python中一种内置的数据结构。元组由不同的元素组成，每个元素可以存储不同类型的数据，如字符串、数字甚至元素。元组是写保护的，即元组

创建之后不能再修改。元组往往代表一行数据，而元组中的元素代表不同的数据项。可以把元组看做不可修改的数组。创建元组示例如下：

```
1. tuple_name=  
    ( "apple" , " banana" , " grape"  
      , " orange" )
```

2 列表 (list)：列表和元组相似，也由一组元素组成，列表可以实现添加、删除和查找操作，元素的值可以被修改。列表是传统意义上的数组。列表创建示例如下：

```
1. list=  
    [ "apple" , " banana" , " grage"  
      , " orange" ]
```

可以使用append方法来在尾部追加元素，使用remove来删除元素。

3 字典 (dictionary)：由键-值对组成的集合，字典中的值通过键来引用。键和值之间用冒号隔开，键-值对之间用逗号隔开，并且被包含在一对花括号中。创建示例如下：

```
1. dict={ "a" : " apple" ,  
          "b" : " banana" ,  
          "g" : " grage" ,  
          "o" : " orange" }
```

4 序列：序列是具有索引和切片能力的集合。元组、列表和字符串都属于序列。

### 【函数相关】

1 python程序由包（package）、模块（module）和函数组成。包是由一系列模块组成的集合。模块是处理某一类问题的函数和类的集合。

2 包就是一个完成特定任务的工具箱。

3 包必须含有一个\_\_init\_\_.py文件，它用于标识当前文件夹是一个包。

4 python的程序是由一个个模块组成的。模块把一组相关的函数或代码组织到一个文件中，一个文件即是一个模块。模块由代码、函数和类组成。导入模块使用import语句。

5 包的作用是实现程序的重用。

6 函数是一段可以重复多次调用的代码，函数定义示例如下：

```
1. def arithmetic(x, y, operator):  
2.     result={  
3.         "+" :x+y,  
4.         "-" :x-y,  
5.         "*" :x*y,  
6.         "/" :x/y  
7.     }
```

7 函数返回值可以用return来控制。

### 【字符串相关】

1 格式化输出：

```
1. format=" %s%d" % (str1, num)  
2. print format
```

2 用+进行字符串的合并：

```
1. str1=" hello"  
2. str2=" world"
```

### 3. result=str1+str2

3 字符串截取可以通过索引/切片，也可以通过split函数。

4 通过切片截取字符串：

1. word=" world"
2. print word[0:3]

5 python使用==和!=来进行字符串比较。如果比较的两个变量的类型不相同，那么结果必然为不同。

### 【文件处理】

1 简单处理文件：

1. context=" hello,world"
2. f=file( "hello.txt", ' w' )
3. f.write(context);
4. f.close()

2 读取文件可以使用readline()函数、readlines()函数和read函数。

3 写入文件可以使用write()、writelines()函数



## 【对象和类】

1 python用class保留字来定义一个类，类名的首字符要大写。当程序员需要创建的类型不能用简单类型来表示时，就需要定义类，然后利用定义的类创建对象。定义类示例：

```
1. class Fruit:  
2.     def grow(self):  
3.         print "Fruit grow"
```

2 当一个对象被创建后，包含了三方面的特性，即对象的句柄、属性和方法。创建对象的方法：

```
1. fruit = Fruit()  
2. fruit.grow()
```

3 python没有保护类型的修饰符

4 类的方法也分为公有方法和私有方法。私有函数不能被该类之外的函数调用，私有的方法也不能被外部的类或函数调用。

5 python使用函数”staticmethod() “或”@

staticmethod “指令的方法把普通的函数转换为静态方法。静态方法相当于全局函数。

6 python的构造函数名为\_\_init\_\_，析构函数名为\_\_del\_\_

7 继承的使用方法：

```
1. class Apple(Fruit):
2. def ...
```

### 【连接mysql】

1 用MySQLdb模块操作MySQL数据库非常方便。示例代码如下：

```
1. import os, sys
2. import MySQLdb
3. try:
4. conn
   MySQLdb.connect(host='localhost',
                    user='root', passwd='',
                    db='address')
5. except Exception, e:
6. print e
7. sys.exit()
```

```
8. cursor=conn.cursor()
9. sql=' insert into
   address(name, address)
   values(%s, %s) '
10. value=
    (( "zhangsan" , " haidian" ),
     ( "lisi" , " haidian" ))
11. try
12. cursor.executemany(sql, values)

13. except Exception, e:
14. print e
15. sql=" select * from address"
16. cursor.execute(sql)
17. data=cursor.fetchall()
18. if data
19. for x in data:
20. print x[0],x[1]
21. cursor.close()
22. conn.close()
```

谢谢!