

# Python程序设计

# 0 简介

是一种[面向对象](#)、[解释型](#)[计算机程序设计语言](#)。

**Python**语法简洁而清楚，具有丰富和强大的类库。

它常被昵称为[胶水语言](#)，能够把用其他语言制作的多种模块（尤其是[C/C++](#)）很轻松地联结在一起。常见的一种应用情形是，使用Python迅速生成[程序](#)的原型（有时甚至是程序的最终界面），然后对其中有尤其要求的部分，用更合适的语言改写，例如[3D游戏](#)中的图形渲染模块，性能要求尤其高，就能够用C/C++重写，而后封装为Python能够调用的扩展类库。需要注意的是在您使用扩展类库时可能需要考虑平台问题，某些可能不提供跨平台的实现。

# 一般用途

**系统编程：**提供API（Application Programming Interface应用程序编程接口），能以便进行系统维护和管理，Linux下标志性语言之一，是诸多系统管理员理想的编程工具。

**图形处理：**有PIL、Tkinter等图形库支持，能以便进行图形处理。

**数学处理：**NumPy扩展提供大量与许多原则数学库的接口。

**文本处理：**python提供的re模块能支持正则体现式，还提供SGML，XML分析模块，许多程序员利用python进行XML程序的开发。

**数据库编程：**程序员可经过遵照Python DB-API（数据库应用程序编程接口）规范的模块与Microsoft SQL Server，Oracle，Sybase，DB2，MySQL、SQLite等数据库通信。python自带有一种Gadfly模块，提供了一种完整的SQL环境。

**网络编程：**提供丰富的模块支持sockets编程，能以便迅速地开发分布式应用程序。诸多大规模软件开发计划例如Zope，Mnet及BitTorrent. Google都在广泛地使用它。

**Web编程：**应用的开发语言，支持最新的XML技术。

**多媒体应用：**Python的PyOpenGL模块封装了“OpenGL应用程序编程接口”，能进行二维和三维图像处理。PyGame模块可用于编写游戏软件。

**pymo引擎：**PYMO全称为python memories off，是一款运营于Symbian S60V3,Symbian3,S60V5, Symbian3, Android系统上的AVG游戏引擎。因其基于python2.0平台开发，而且合用于创建秋之回忆（memories off）风格的AVG游戏，故命名为PYMO。

**黑客编程：**python有一种hack的库,内置了你熟悉的或不熟悉的函数，但是缺乏成就感。

python目前主要在做**web**站点（php此前做的工作，例子：豆瓣，知乎）服务器端的服务和系统（比java轻量级，开发迅速，例子，youtube，dropbox，openstack），**大数据**中的数据处理（报表，mapreduce）。目前看python的主要方向是取代php以及java在web服务器端的部分份额，以及在海量数据系统中起到处理数据的脚本的功能。

# 1 安装

**下载:** 32位系统选“Python 2.7.5 Windows Installer”，假如是64位系统的同学选下面那个“Python 2.7.5 Windows X86-64 Installer”

**设置环境变量:** 右键单击我的电脑，依次点击“属性”->“高级”->“环境变量”，在“系统变量”表单中点击叫做Path的变量，然后编辑这个变量，把“;C:\Python27\”，也就是你刚刚复制的安装途径，加到它的结尾。

## 2. print

`print`，中文意思是打印，在python里它是打印在命令行，或者叫终端、控制台里面。

`print`是python里很基本很常见的一种操作，它的操作对象是一种字符串。

基本格式是：`print` 你要打印的东西或者 `print(你要打印的东西)`这里一定要英文字符的括号。直接在`print`背面加一段文字来输出的话，需要给文字加上双引号或者单引号。

`print`除了打印文字之外，还能输出多种数字、运算成果、比较成果等。

`python`命令行下，`print`是能够省略的，默认就会输出每一次命令的成果。

# 3 IDE

python自帶了一款IDE，叫做IDLE。Windows上安裝了之後，能够在“开始菜单” -> “程序” -> “Python 2.7” 里找到它。

点击窗口上方菜单栏的“File”->“New Window”，会打一种长得很像的新窗口，但里面什么也没有。这是一种文本编辑器，在这里面就能够写我们的python程序了。继续print几行，这次能够多print一点：

```
print 'Hello'  
print 'IDE'  
print 'Here I am.'
```

点击“Run”->“Run Module”，或者直接按快捷键F5。会提醒你保存刚刚文件，随便取个名字，例如“333.py”。（.py是python代码文件的类型，虽然不指定.py也是能够的，但提议还按规范来）保存完毕后，之前那个控制台窗口里就会一次性输出你要的成果。

后来想再次编辑或运营刚刚的代码，只要在IDLE里选择“File”->“Open...”，打开刚刚保存的.py文件就足够了。

# 4 输入

## **input()**

注意，和print不同的是，这次我们必须得加上()了，而且得是英文字符的括号。

**raw\_input()**，它把全部的输入都直接看成一串字符

# 5 变量

变量，望文生义，就是变化的量。python里创建一种变量的措施很简朴.

```
name = 'Crossin'
```

```
myVar = 123
```

```
price = 5.99
```

```
visible = True
```

“=”的作用是把右边的值赋予给左边的变量。

用print输出一种“变量”：

```
name = 'Crossin'
```

```
print name
```

它叫变量，那就是能变的。所以在一次“赋值”操作之后，还能够继续给它赋予新的值，而且能够是不同类型的值。

```
a = 123
```

```
print a
```

```
a = 'hi'
```

```
print a
```

“=” 的右边还能够更复杂一点，例如是一种计算出昀值：

```
value = 3 * 4
```

```
print value
```

```
value = 2 < 5
```

```
print value
```

甚至，也能够是input():

```
name = input()  
print name
```

```
print "Who do you think I am?"  
you = input()  
print "Oh, yes! I am a"  
print you
```

# 6 bool

- 逻辑判断在编程中是非常重要的。大量的复杂程序在根本上都是建立在“真”与“假”的基本逻辑之上。而bool所表达的就是这种最单纯最本质的True / Flase，真与假，是与非。

>: 不小于

<: 不不小于

>=: 不小于等于

<=: 不不小于等于

==: 等于。比较两个值是否相等。之所以用两个等号，是为了和变量赋值区别开来。

!=: 不等与

not: 逻辑“非”。假如x为True，则not x为False

and: 逻辑“与”。假如x为True，且y为True，  
则x and y为True

or: 逻辑“或”。假如x、y中至少有一种为True，  
则x or y为True

```
num = 10  
print 'Guess what I think?'  
answer = input()
```

```
result = answer < num  
print 'too small?'  
print result
```

```
result = answer > num  
print 'too big?'  
print result
```

```
result = answer == num  
print 'equal?'  
print result
```

# 7 if

if在编程语言中被称为“控制流语句”，用来控制程序的执行顺序  
“假如”为True，就会去执行接下来的内容。“假如”为False，就跳过

if 条件:

选择执行的语句

尤其阐明：条件背面的**冒号不能少**，一样必须是英文字符。

尤其尤其阐明：if内部的语句需要有一种**统一的缩进**，一般用4个空格。

python用这种措施替代了其他诸多编程语言中的{}。你也能够选择1/2/3...个空格或者按一下tab键，但必须整个文件中都统一起来。千万不能够tab和空格混用，不然就会出现多种莫名其妙的错误。所以提议都直接用4个空格。

```
thisLove = input()
```

```
if thisLove:
```

```
    print "再转身就该勇敢留下来"
```

```
num = 10
print 'Guess what I think?'
answer = input()
if answer < num:
    print 'too small!'

if answer > num:
    print 'too big!'
if answer == num:
    print 'BINGO!'
```

# 8 while

同if一样，**while**也是一种控制流语句，另外它也被称作循环语句。

程序执行到**while**处，“当”条件为True时，就去执行**while**内部的代码，“当”条件为False时，就跳过。

语法为：

**while** 条件:

    循环执行的语句

同if一样，注意冒号，注意缩进。

# 9 random

之前我们用了诸屡次的print和input措施，它们的作用是实现控制台的输入和输出。除此之外，python还提供了诸多模块，用来实现多种常见的功能，例如时间处理、科学计算、网络祈求、随机数等等等等。今日我就来说说，怎样用python自带的随机数模块。

引入模块的措施：

**from 模块名 import 措施名**

```
from random import randint
```

之后你就能够用randint来产生随机数了。

```
randint(5, 10)
```

这么将会产生一种5到10之间（涉及5和10）的随机整数。

```
answer = randint(1, 100)
```

程序在运营时候，会产生一种1到100的随机整数，存在answer里。

# 10 for循环

```
for ... in ...
```

```
for i in range(1, 101):  
    print i
```

解释一下，`range(1, 101)`表达从1开始，到101为止（不涉及101），取其中全部的整数。`for i in range(1, 101)`就是说，把这些数，依次赋值给变量*i*。相当于一种一种循环过去，第一次*i* = 1，第二次*i* = 2，……，直到*i* = 100。当*i* = 101时跳出循环。

# 11 字符串

python中最常用的字符串表达方式是单引号（`'`）和双引号（`"`）。

能够直接输出一种字符串，也能够用一种变量来保存字符串，然后输出

python中还有一种表达字符串的措施：三个引号（`'''`）或者（`"""`）

字符串中表达引号也能够用`\`，能够不受引号的限制（转义字符）

\'表达单引号， \"表达双引号  
'\`m a `good` teacher'

\被称作转译字符，除了用来表达引号，还有  
例如用

\\表达字符串中的\  
\n表达字符串中的换行

\还有个用处，就是用来在代码中换行，而不  
影响输出的成果：

"this is the\  
same line"

# 作业

作业：用print输出下列文字：

1. He said, "I'm yours!"

2. \\\_v\_//

3. Stay hungry,  
stay foolish.

-- Steve Jobs

4.

\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*

# 12 字符串格式化

输出字符串的时候，假如想对输出的内容进行某些整顿，例如把几段字符拼接起来，或者把一段字符插入到另一段字符中间，就需要用到字符串的格式化输出。

**把两段字符连起来输出**

```
str1 = 'good'
```

```
str2 = 'bye'
```

```
print str1 + str2
```

## 一种数字加到文字背面输出

1. 用str()把数字转换成字符串

```
print 'My age is' + str(18)
```

或

```
num = 18
```

```
print 'My age is' + str(num)
```

2. 就是用%对字符串进行格式化

```
num = 18
```

```
print 'My age is %d' % num
```

输出的时候，%d会被%背面的值替代。输出

```
My age is 18
```

%d只能用来替代整数。格式化的数值是小数，  
要用%f

```
print 'Price is %f' % 4.99
```

假如你想保存两位小数，需要在f前面加上条件：  
%.2f

```
print 'Price is %.2f' % 4.99
```

能够用%s来替代一段字符串

```
name = 'Crossin'
```

```
print '%s is a good teacher.' % name
```

```
print 'Today is %s.' % 'Friday'
```

注意区别：有引号的表达一段字符，没有引号的就是一种变量，这个变量可能是字符，也可能是数字，但一定要和%所示的格式相一致。

设你目前有一组学生成绩的数据，你要输出这些数据。在一行中，既要输出学生的姓名，又要输出他的成绩。例如

Mike's score is 87.

```
print "%s's score is %d" % ('Mike', 87)
```

```
name = 'Lily'
```

```
score = 95
```

```
print "%s's score is %d" % (name, score)
```

不论你有多少个值需要代入字符串中进行格式化，只需要在字符串中的合适位置用相应格式的%表达，然后在背面的括号中按顺序提供代入的值就足够了。占位的%和括号中的值在数量上必须相等，类型也要匹配。

# 13 循环的嵌套

```
for i in range(0, 5):  
    for j in range(0, 5):  
        print i, j
```

# 14 类型转换

`int(x)` #把x转换成整数

`float(x)` #把x转换成浮点数

`str(x)` #把x转换成字符串

`bool(x)` #把x转换成bool值

因为在python中，下列数值会被以为是False：

为0的数字，涉及0，0.0

空字符串，涉及"，""

表达空值的None

空集合，涉及()，[]，{}

其他的值都以为是True。

None是python中的一种特殊值，表达什么都没有，它和0、空字符、False、空集合都不同

。

# 15 函数

自己写一种函数，就需要去 定义 它。

python里的关键字叫def（define的缩写），格式如下：

```
def sayHello():  
    print 'hello world!'
```

sayHello是这个函数的名字，背面的括号里是参数，这里没有，表达不需要参数。但括号和背面的冒号都不能少。下面缩进的代码块就是整个函数的内容，称作函数体。

```
def sayHello(someone):  
    print someone + ' says Hello!'
```

```
def plus(num1, num2):  
    print num1+num2
```

```
def plus(num1, num2):  
    return num1+num2
```

# 16 if, elif, else

```
if a == 1:  
    print 'one'  
elif a == 2:  
    print 'two'  
elif a == 3:  
    print 'three'  
else:  
    print 'too many'
```

# 17 if的嵌套

if 条件1:

    if 条件2:

        语句1

    else:

        语句2

else:

    if 条件2:

        语句3

    else:

        语句4

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/335210044241011341>