

## Python 入门教程 超详细 1 小时学会 Python

本文适合有经验的程序员尽快进入 [Python](#) 世界. 特别地, 如果你掌握 [Java](#) 和 Javascript, 不用 1 小时你就可以用 Python 快速流畅地写有用的 Python 程序.

### 为什么使用 Python

假设我们有这么一项任务: 简单测试局域网中的电脑是否连通. 这些电脑的 ip 范围从 192. 168. 0. 101 到 192. 168. 0. 200.

思路: 用 shell 编程. ([Linux](#) 通常是 bash 而 [Windows](#) 是批处理脚本). 例如, 在 Windows 上用 ping ip 的命令依次测试各个机器并得到控制台输出. 由于 ping 通的时候控制台文本通常是 "Reply from ..." 而不通的时候文本是 "time out ... ", 所以, 在结果中进行字符串查找, 即可知道该机器是否连通.

实现: Java 代码如下:

```
String cmd="cmd.exe ping ";
String ipprefix="192.168.10.";
int begin=101;
int end=200;
Process p=null;
for(int i=begin;i<end;i++){
    p= Runtime.getRuntime().exec(cmd+i);
    String line = null;
    BufferedReader reader = new BufferedReader(new
        InputStreamReader(p.getInputStream()));
    while((line = reader.readLine()) != null)
    {
        //Handling line , may logs it.
    }
    reader.close();
    p.destroy();
}
```

这段代码运行得很好, 问题是为了运行这段代码, 你还需要做一些额外的工作. 这些额外的工作包括:

编写一个类文件

编写一个 main 方法

将之编译成字节代码

由于字节代码不能直接运行, 你需要再写个小小的 bat 或者 bash 脚本来运行.

当然, 用 C/C++ 同样能完成这项工作. 但 C/C++ 不是跨平台语言. 在这个足够简单的例子中也许看不出 C/C++ 和 Java 实现的区别, 但在一些更为复杂的场景, 比如

要将连通与否的信息记录到网络数据库. 由于 Linux 和 Windows 的网络接口实现方式不同, 你不得不写两个函数的版本. 用 Java 就没有这样的顾虑. 同样的工作用 Python 实现如下:

```
import subprocess
cmd="cmd.exe"
begin=101
end=200
while begin<end:
p=subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE,
stdin=subprocess.PIPE,
stderr=subprocess.PIPE)
p.stdin.write("ping 192.168.1."+str(begin)+"\n")
p.stdin.close()
p.wait()
print "execution result: %s"%p.stdout.read()
```

对比 Java, Python 的实现更为简洁, 你编写的时间更快. 你不需要写 main 函数, 并且这个程序保存之后可以直接运行. 另外, 和 Java 一样, Python 也是跨平台的. 有经验的 C/Java 程序员可能会争论说用 C/Java 写会比 Python 写得快. 这个观点见仁见智. 我的想法是当你同时掌握 Java 和 Python 之后, 你会发现用 Python 写这类程序的速度会比 Java 快上许多. 例如操作本地文件时你仅需要一行代码而不需要 Java 的许多流包装类. 各种语言有其天然的适合的应用范围. 用 Python 处理一些简短程序类似与操作系统的交互编程工作最省时省力.

---

## Python 应用场合

足够简单的任务, 例如一些 shell 编程. 如果你喜欢用 Python 设计大型商业网站或者设计复杂的游戏, 悉听尊便.

---

## 2 快速入门 2.1 Hello world

安装完 Python 之后 (我本机的版本是 2.5.4), 打开 IDLE (Python GUI), 该程序是 Python 语言解释器, 你写的语句能够立即运行. 我们写下一句著名的程序语句:

```
print "Hello, world!"
```

并按回车. 你就能看到这句被 K&R 引入到程序世界的名言.

在解释器中选择 "File" -- "New Window" 或快捷键 Ctrl+N, 打开一个新的编辑器. 写下如下语句:

```
print "Hello, world!"
```

```
raw_input("Press enter key to close this window");
```

保存为 a.py 文件. 按 F5, 你就可以看到程序的运行结果了. 这是 Python 的第二种运行方式.

找到你保存的 a.py 文件, 双击. 也可以看到程序结果. Python 的程序能够直接运行, 对比 Java, 这是一个优势.

---

## 2.2 国际化支持

我们换一种方式来问候世界. 新建一个编辑器并写如下代码:

```
print "欢迎来到奥运中国!"
```

```
raw_input("Press enter key to close this window");
```

在你保存代码的时候, Python 会提示你是否改变文件的字符集, 结果如下:

```
# -*- coding: cp936 -*-
```

```
print "欢迎来到奥运中国!"
```

```
raw_input("Press enter key to close this window");
```

将该字符集改为我们更熟悉的形式:

```
# -*- coding: GBK -*-
```

```
print "欢迎来到奥运中国!" # 使用中文的例子
```

```
raw_input("Press enter key to close this window");
```

程序一样运行良好.

---

## 2.3 方便易用的计算器

用微软附带的计算器来计数实在太麻烦了. 打开 Python 解释器, 直接进行计算:

```
a=100.0
```

```
b=201.1
```

```
c=2343
```

```
print (a+b+c)/c
```

---

## 2.4 字符串, ASCII 和 UNICODE

可以如下打印出预定义输出格式的字符串:

```
print """
```

```
Usage: thingy [OPTIONS]
```

```
-h Display this usage message
```

```
-H hostname Hostname to connect to
```

```
"""
```

字符串是怎么访问的? 请看这个例子:

```
word="abcdefg"
```

```
a=word[2]
```

```
print "a is: "+a
```

```
b=word[1:3]
```

```
print "b is: "+b # index 1 and 2 elements of word.
```

```

c=word[:2]
print "c is: "+c # index 0 and 1 elements of word.
d=word[0:]
print "d is: "+d # All elements of word.
e=word[:2]+word[2:]
print "e is: "+e # All elements of word.
f=word[-1]
print "f is: "+f # The last elements of word.
g=word[-4:-2]
print "g is: "+g # index 3 and 4 elements of word.
h=word[-2:]
print "h is: "+h # The last two elements.
i=word[:-2]
print "i is: "+i # Everything except the last two characters
l=len(word)
print "Length of word is: "+ str(l)
请注意 ASCII 和 UNICODE 字符串的区别:
print "Input your Chinese name:"
s=raw_input("Press enter to be continued");
print "Your name is : " +s;
l=len(s)
print "Length of your Chinese name in asc codes is:"+str(l);
a=unicode(s,"GBK")
l=len(a)
print "I'm sorry we should use unicode char!Characters number of your
Chinese \
name in unicode is:"+str(l);

```

---

## 2.5 使用 List

类似 Java 里的 List, 这是一种方便易用的数据类型:

```

word=['a','b','c','d','e','f','g']
a=word[2]
print "a is: "+a
b=word[1:3]
print "b is: "
print b # index 1 and 2 elements of word.
c=word[:2]
print "c is: "
print c # index 0 and 1 elements of word.
d=word[0:]
print "d is: "
print d # All elements of word.

```

```

e=word[:2]+word[2:]
print "e is: "
print e # All elements of word.
f=word[-1]
print "f is: "
print f # The last elements of word.
g=word[-4:-2]
print "g is: "
print g # index 3 and 4 elements of word.
h=word[-2:]
print "h is: "
print h # The last two elements.
i=word[:-2]
print "i is: "
print i # Everything except the last two characters
l=len(word)
print "Length of word is: "+ str(l)
print "Adds new element"
word.append('h')
print word

```

---

## 2.6 条件和循环语句

```

# Multi-way decision
x=int(raw_input("Please enter an integer:"))
if x<0:
    x=0
print "Negative changed to zero"
elif x==0:
    print "Zero"
else:
    print "More"

# Loops List
a = ['cat', 'window', 'defenestrate']
for x in a:
    print x, len(x)

```

---

## 2.7 如何定义函数

```

# Define and invoke function.
def sum(a,b):
    return a+b

```

```

func = sum
r = func(5,6)
print r
# Defines function with default argument
def add(a,b=2):
    return a+b
r=add(1)
print r
r=add(1,5)
print r
并且, 介绍一个方便好用的函数:
# The range() function
a =range(5,10)
print a
a = range(-2,-7)
print a
a = range(-7,-2)
print a
a = range(-2,-11,-3) # The 3rd parameter stands for step
print a

```

---

## 2.8 文件 I/O

```

spath="D:/download/baa.txt"
f=open(spath,"w") # Opens file for writing. Creates this file doesn't
exist.
f.write("First line 1.\n")
f.writelines("First line 2.")
f.close()
f=open(spath,"r") # Opens file for reading
for line in f:
    print line
f.close()

```

---

## 2.9 异常处理

```

s=raw_input("Input your age:")
if s == "":
    raise Exception("Input must no be empty.")
try:
    i=int(s)
except ValueError:

```

```

print "Could not convert data to an integer."
except:
print "Unknown exception!"
else: # It is useful for code that must be executed if the try clause does
not raise an exception
print "You are %d" % i, " years old"
finally: # Clean up action
print "Goodbye!"

```

---

## 2.10 类和继承

```

class Base:
def __init__(self):
self.data = []
def add(self, x):
self.data.append(x)
def addtwice(self, x):
self.add(x)
self.add(x)
# Child extends Base
class Child(Base):
def plus(self, a, b):
return a+b
oChild =Child()
oChild.add("str1")
print oChild.data
print oChild.plus(2, 3)

```

---



---

## 2.11 包机制

每一个.py 文件称为一个 module,module 之间可以互相导入. 请参看以下例子:

```

# a.py
def add_func(a, b):
return a+b
# b.py
from a import add_func # Also can be : import a
print "Import add_func from module a"
print "Result of 1 plus 2 is: "
print add_func(1,2) # If using "import a" , then here should be
"a.add_func"

```

module 可以定义在包里面. Python 定义包的方式稍微有点古怪, 假设我们有一个

parent 文件夹, 该文件夹有一个 child 子文件夹. child 中有一个 module a.py . 如何让 Python 知道这个文件层次结构?很简单, 每个目录都放一个名为 `__init__.py` 的文件. 该文件内容可以为空. 这个层次结构如下所示: parent

```
--__init__.py
--child
-- __init__.py
--a.py
b.py
```

那么 Python 如何找到我们定义的 module?在标准包 `sys` 中, `path` 属性记录了 Python 的包路径. 你可以将之打印出来:

```
import sys
print sys.path
```

通常我们可以将 module 的包路径放到环境变量 `PYTHONPATH` 中, 该环境变量会自动添加到 `sys.path` 属性. 另一种方便的方法是编程中直接指定我们的 module 路径到 `sys.path` 中:

```
import sys
sys.path.append('D:\\download')
from parent.child.a import add_func

print sys.path
print "Import add_func from module a"
print "Result of 1 plus 2 is: "
print add_func(1,2)
```

---

总结你会发现这个教程相当的简单. 许多 Python 特性在代码中以隐含方式提出, 这些特性包括: Python 不需要显式声明数据类型, 关键字说明, 字符串函数的解释等等. 我认为一个熟练的程序员应该对这些概念相当了解, 这样在你挤出宝贵的一小时阅读这篇短短的教程之后, 你能够通过已有知识的迁移类比尽快熟悉 Python, 然后尽快能用它开始编程.



常用的 python 模块功能和下载地址

2009-07-13 11:09

adodb: 我们领导推荐的[数据库](#)连接组件

bsddb3: BerkeleyDB 的连接组件

Cheetah-1.0: 我比较喜欢这个版本的 cheetah

cherrypy: 一个 [WEB](#) framework

ctypes: 用来调用动态链接库

DBUtils: 数据库连接池

django: 一个 WEB framework

docutils: 用来写文档的

dpkt: 数据包的解包和组包

MySQLdb: 连接 [MySQL](#) 数据库的

py2exe: 用来生成 [windows](#) 可执行文件

Pylons: 我们领导推荐的 web framework

pysqlite2: SQLite 的连接组件

pythonwin: Python 的 Windows 扩展

setuptools: 无奈, PEAK 的一套 python 包管理机制

sqlalchemy: 数据库连接池

SQLObject: 数据库连接池

twisted: 巨无霸的网络编程框架

wxPython-2.6: 因为需要保持与 linux 相同的版本才没用最新的, GUI 编程框架

pypcap: 抓包的

python-dnet: 控制网络安全的[其他](#)设备

pyevent: Python 的事件支持

pydot: 画图的, graphviz

sendpkt: Python 发包

simplejson: JSON 的支持

DPKT: raw-socket 网络编程

Cx-oracle: 连接 oracle 的好东东

Mechanize: 爬虫连接网站常用

PIL: 图像处理工具包

reportlab for PDF 文件。

PyQt4 for GUI 界面

feedparser: rss 解析

chardet: 编码检测

scons: 项目构建工具, 写好了模板用起来还是很方便的

scapy: 网络包构建分析框架, 可编程的 wireshark, 有兴趣的 google “Silver Needle in the Skype”

pefile: windows pe 文件解析器

winpdb: 自己的程序或者用别的库不太明白的时候就靠它了  
pywmi: 省了好多折腾功夫  
pylint: 培养良好的编码习惯

下面是准备用的, 不做评论:

pygccxml  
pyparsing  
pymacs  
idapython  
paimei  
pysvn  
pyLucene  
wikidpad

1. Python 操作 PDF 文件
2. Python 连接 PostgreSQL 引擎
3. Python 的 .NET2.0 的曲线表

<http://pybrary.net/>

---

1. Python Image Library
2. Tkinter Library

<http://www.pythonware.com/index.htm>

---

1. Karrigell

<http://karrigell.sourceforge.net/>

---

1. The Standard Python Library

<http://effbot.org/librarybook/>

---

1. Python 操作 PDF 的 Library

<http://reportlab.org/>

---

1. Python cURL Library

<http://pycurl.sourceforge.net/>

---

1. urlgrabber

<http://linux.duke.edu/projects/urlgrabber/>

---

#### 1. MySQL-Python

<http://mysql-python.sourceforge.net/>

---

#### 1. pyscripter

#### 2. Python for Delphi

<http://mmm-experts.com/Products.aspx>

---

#### 1. Python Cookbook

<http://aspn.activestate.com/ASPN/Python/Cookbook/>

---

#### 1. pygame

<http://www.pygame.org/news.html>

---

#### 1. pymedia

<http://www.pymedia.org>

---

#### 1. pydbf

<http://dbfpy.sourceforge.net/>

---

#### 1. Pexpect

<http://pexpect.sourceforge.net/>

---

#### 1. dabo

desktop application frameworks

<http://dabodev.com/>

---

#### 1. pySerial

<http://pyserial.sourceforge.net/>

<http://wiki.woodpecker.org.cn/moin/PySerial>

---

#### 1. python' s doc

<http://wiki.woodpecker.org.cn/moin/PythonZhDoc>

---

1. The Visual Module of VPython(3D)

<http://vpython.org/index.html>

---

1. PyLucene

<http://pylucene.osafoundation.org/>

---

1. DBUtils

<http://www.webwareforpython.org/DBUtils>

---

1. <http://happyiww.popo.blog.163.com/blog/>

a good people' s blog

---

1. Applescript, OSAComponents, HtmlCalendar

<http://freespace.virgin.net/hamish.sanderson/index.html>

---

1. Pypcap

<http://code.google.com/p/pypcap/>

<http://monkey.org/~dugsong/dpkt/>

<http://www.winpcap.org/install/default.htm>

python-memcached client module for memcached

Kodos, python 的正则表达式调试工具,

PLY (Python Lex-Yacc)

<http://www.dabeaz.com/ply/>

正在用它写汇编器.

wxPython - GUI Framework

omniORBpy - CORBA Library

ZODB3 - an OODB implement

ReportLab - PDF generating solution

numpy+scipy+matplotlib

hachoir 各种多媒体元数据解析  
python-orge 演示用

数据库的:

sqlalchemy

sqlobject

MySQL-python (easy\_install 的时候就是这个名字)

routes

beaker

两个模板引擎:

genshi

mako

paster

另外还有知道的

PyCrypto 加密的

paramiko 这个可以帮你写 sftp, 比 twisted 里面的那个好用一些

twisted 写服务器玩的

PyOpenGL

PyGame

Psyco 传说可以加速你的 python 程序

PIL (Python Image Library) 处理图片

PyObjC 这个是苹果上用的, 写 Cocoa 程序

Python desktop client GUI application:

wxPython

PyOpenGL

PyXML