



哥伦布的小寨子

glbfor.blog.chinaunix.net

what can I do is not about your words, but what I have done is absolutely about your opinions, just so. So ganbaru!

[首页](#) | [博文目录](#) | [关于我](#)

glbfor

博客访问： 49927
博文数量： 27
博客积分： 414
博客等级： 下士
技术积分： 247
用户组： 普通用户
注册时间： 2011-10-06 15:52

[加关注](#) [短消息](#)
[论坛](#) [加好友](#)

个人简介

身为挨踢人，却不想被挨踢。努力多学点。

文章分类

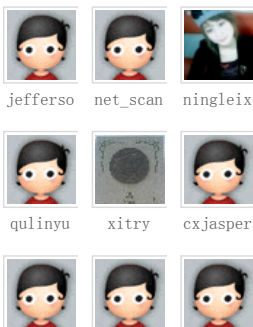
全部博文 (27)
 杂乱 (5)
 linux_cmd (1)
 未分配的博文 (21)

文章存档

2012年 (17)
2011年 (10)

我的朋友

最近访客



python常用的一些东西——sys、os等(转)

2012-09-19 14:51:49

分类：

原文地址：python常用的一些东西——sys、os等(转) 作者：peijieking

1. 常用内置函数：（不用import就可以直接使用）

help(obj) 在线帮助，obj可是任何类型
callable(obj) 查看一个obj是不是可以像函数一样调用
repr(obj) 得到obj的表示字符串，可以利用这个字符串eval重建该对象的一个拷贝
eval_r(str) 表示合法的python表达式，返回这个表达式
dir(obj) 查看obj的name space中可见的name
hasattr(obj, name) 查看一个obj的name space中是否有name
getattr(obj, name) 得到一个obj的name space中的一个name
setattr(obj, name, value) 为一个obj的name space中的一个name指向vale这个object
delattr(obj, name) 从obj的name space中删除一个name
vars(obj) 返回一个object的name space。用dictionary表示
locals() 返回一个局部name space, 用dictionary表示
globals() 返回一个全局name space, 用dictionary表示
type(obj) 查看一个obj的类型
isinstance(obj, cls) 查看obj是不是cls的instance
issubclass(subcls, supcls) 查看subcls是不是supcls的子类

类型转换函数

chr(i) 把一个ASCII数值, 变成字符
ord(i) 把一个字符或者unicode字符, 变成ASCII数值
oct(x) 把整数x变成八进制表示的字符串
hex(x) 把整数x变成十六进制表示的字符串
str(obj) 得到obj的字符串描述
list(seq) 把一个sequence转换成一个list
tuple(seq) 把一个sequence转换成一个tuple
dict(), dict(list) 转换成一个dictionary
int(x) 转换成一个integer
long(x) 转换成一个long interger
float(x) 转换成一个浮点数
complex(x) 转换成复数
max(...) 求最大值
min(...) 求最小值

用于执行程序的内置函数

compleie 如果一段代码经常要使用, 那么先编译, 再运行会更快。

2. 和操作系统相关的调用

系统相关的信息模块 import sys

sys.argv是一个list, 包含所有的命令行参数。

sys.stdout sys.stdin sys.stderr 分别表示标准输入输出, 错误输出的文件对象。

sys.stdin.readline() 从标准输入读一行 sys.stdout.write("a") 屏幕输出a

sys.exit(exit_code) 退出程序

xiaoxiao wang was 泰天道人

微信关注



IT168企业级官微

微信号: IT168qiye



系统架构师大会

微信号: SACC2013

订阅

推荐博文

- Docker+Redis3集群环境搭建...
- LVM 类型的 Storage Pool - ...
- 自动化运维之cobbler批量部署...
- Redhat下puppet集中配置管理...
- 使用flume+kafka+storm构建实...
- MySQL数据碎片的整理和分析...
- Web报表工具FineReport的JS A...
- oracle 12c 数据归档 即Using...
- Oracle手动固定SQL执行计划...
- 11gR2 GI和DB安装目录权限属...

热词专题

- ubuntu 格式化U盘
- 清理ORACLE归档日志
- 多显示器的显示
- linux命令
- GDB

sys.modules 是一个dictionary, 表示系统中所有可用的module
 sys.platform 得到运行的操作系统环境
 sys.path 是一个list, 指明所有查找module, package的路径.

操作系统相关的调用和操作 import os

os.environ 一个dictionary 包含环境变量的映射关系 os.environ["HOME"] 可以得到环境变量HOME的值

os.chdir(dir) 改变当前目录 os.chdir('d:\\outlook') 注意windows下用到转义

os.getcwd() 得到当前目录

os.getegid() 得到有效组id os.getgid() 得到组id

os.getuid() 得到用户id os.geteuid() 得到有效用户id

os.setegid os.setegid() os.seteuid() os.setuid()

os.getgruops() 得到用户组名称列表

os.getlogin() 得到用户登录名称

os.getenv 得到环境变量

os.putenv 设置环境变量

os.umask 设置umask

os.system(cmd) 利用系统调用, 运行cmd命令

操作举例:

```
os.mkdir('/tmp/xx') os.system("echo 'hello' > /tmp/xx/a.txt")
```

```
os.listdir('/tmp/xx')
```

```
os.rename('/tmp/xx/a.txt', '/tmp/xx/b.txt') os.remove('/tmp/xx/b.txt')
```

```
os.rmdir('/tmp/xx')
```

用python编写一个简单的shell

```
#!/usr/bin/python
import os, sys
cmd = sys.stdin.readline()
while cmd:
    os.system(cmd)
    cmd = sys.stdin.readline()
```

用os.path编写平台无关的程序

```
os.path.abspath("l.txt") == os.path.join(os.getcwd(), "l.txt")
```

os.path.split(os.getcwd()) 用于分开一个目录名称中的目录部分和文件名称部分。

os.path.join(os.getcwd(), os.pardir, 'a', 'a.doc') 全成路径名称.

os.pardir 表示当前平台下一级目录的字符 ..

os.path.getctime("/root/l.txt") 返回l.txt的ctime(创建时间)时间戳

os.path.exists(os.getcwd()) 判断文件是否存在

os.path.expanduser('~ /dir') 把~扩展成用户根目录

os.path.expandvars('\$PATH') 扩展环境变量PATH

os.path.isfile(os.getcwd()) 判断是否是文件名, 1是0否

os.path.isdir('c:\Python26\temp') 判断是否是目录, 1是0否

os.path.islink('/home/huaying/111.sql') 是否是符号连接 windows下不可用

os.path.ismount(os.getcwd()) 是否是文件系统安装点 windows下不可用

os.path.samefile(os.getcwd(), '/home/huaying') 看看两个文件名是不是指的是同一个文件

os.path.walk('/home/huaying', test_fun, "a.c")

遍历/home/huaying下所有子目录包括本目录, 对于每个目录都会调用函数test_fun.

例: 在某个目录中, 和他所有的子目录中查找名称是a.c的文件或目录。

```
def test_fun(filename, dirname, names): //filename即是walk中的a.c dirname是访问的目录名称
```

```
if filename in names: //names是一个list, 包含dirname目录下的所有内容
```

```
print os.path.join(dirname, filename)
```

```
os.path.walk('/home/huaying', test_fun, "a.c")
```

文件操作

打开文件

f = open("filename", "r") r只读 w写 rw读写 rb读二进制 wb写二进制 w+写追加

读写文件

f.write("a") f.write(str) 写一字符串 f.writeline() f.readlines() 与下read类同

f.read() 全读出来 f.read(size) 表示从文件中读取size个字符

f.readline() 读一行,到文件结尾,返回空串. f.readlines() 读取全部, 返回一个list.

list每个元素表示一行, 包含"\n"

f.tell() 返回当前文件读取位置

f.seek(off, where) 定位文件读写位置. off表示偏移量, 正数向文件尾移动, 负数表示向开头移动。

where为0表示从开始算起, 1表示从当前位置算, 2表示从结尾算.

f.flush() 刷新缓存

关闭文件

f.close()

regular expression 正则表达式 import re

简单的regexp

p = re.compile("abc") if p.match("abc") : print "match"

上例中首先生成一个pattern(模式), 如果和某个字符串匹配, 就返回一个match object

除某些特殊字符metacharacter元字符, 大多数字符都和自身匹配。

这些特殊字符是 . ^ \$ * + ? { [] \ | ()

字符集合(用[]表示)

列出字符, 如[abc]表示匹配a或b或c, 大多数metacharacter在[]中只表示和本身匹配。例:

a = ".\$*+?{\|() " 大多数metachar在[]中都和本身匹配, 但"^[]\"不同

p = re.compile("[^a+\"")

for i in a:

if p.match(i):

print "[%s] is match" %i

else:

print "[%s] is not match" %i

在[]中包含[]本身, 表示"[\"或者"]"匹配. 用\[和\]表示.

^出现在[]的开头, 表示取反. [^abc]表示除了a, b, c之外的所有字符. ^没有出现在开头, 即于身身匹配。

-可表示范围. [a-zA-Z]匹配任何一个英文字母. [0-9]匹配任何数字。

\在[]中的妙用。

\d [0-9]

\D [^0-9]

\s [\t\n\r\f\v]

\S [^ \t\n\r\f\v]

\w [a-zA-Z0-9_]

\W [^a-zA-Z0-9_]

\t 表示和tab匹配, 其他的都和字符串的表示法一致

\x20 表示和十六进制ascii 0x20匹配

有了\, 可以在[]中表示任何字符。注: 单独的一个"."如果没有出现[]中, 表示出了换行\n以外的匹配任何字符, 类似"[^n]".

regexp的重复

{m, n}表示出现m个以上(含m个), n个以下(含n个). 如ab{1, 3}c和abc, abbc, abbbc匹配, 不会与ac, abbbc匹配。

m是下界, n是上界. m省略表下界是0, n省略, 表上界无限大。

*表示{, } +表示{1, } ?表示{0, 1}

最大匹配和最小匹配 python都是最大匹配, 如果要最小匹配, 在*, +, ?, {m, n}后面加一个?.

match object的end可以得到匹配的最后一个字符的位置。

re.compile("a*").match('aaaa').end() 4 最大匹配

re.compile("a*?").match('aaaa').end() 0 最小匹配

使用原始字符串

字符串表示方法中用\\表示字符\, 大量使用影响可读性。

解决方法：在字符串前面加一个r表示raw格式。

```
a = r"\a" print a 结果是\a
```

```
a = r"\"a" print a 结果是\"a
```

使用re模块

先用re.compile得到一个RegexObject 表示一个regexp

后用pattern的match, search的方法, 得到MatchObject

再用match object得到匹配的位置, 匹配的字符串等信息

RegexObject常用函数:

```
>>> re.compile("a").match("abab") 如果abab的开头和re.compile("a")匹配，得到MatchObject
```

```
<_sre.SRE_Match object at 0x81d43c8>
```

```
>>> print re.compile("a").match("bbab")
```

None 注：从str的开头开始匹配

```
>>> re.compile("a").search("abab") 在abab中搜索第一个和re_obj匹配的部分
```

```
<_sre.SRE_Match object at 0x81d43c8>
```

```
>>> print re.compile("a").search("bbab")
```

```
<_sre.SRE_Match object at 0x8184e18> 和match()不同, 不必从开头匹
```

配

re_obj.findall(str) 返回str中搜索所有和re_obj匹配的部分。

返回一个tuple, 其中元素是匹配的字符串。

MatchObject的常用函数

m.start() 返回起始位置, m.end() 返回结束位置(不包含该位置的字符)。

m.span() 返回一个tuple表示(m.start(), m.end())

m.pos(), m.endpos(), m.re(), m.string()

m.re().search(m.string(), m.pos(), m.endpos()) 会得到m本身

m.finditer() 可以返回一个iterator, 用来遍历所有找到的MatchObject。

```
for m in re.compile("[ab]").finditer("tatbxaxb"):
```

```
print m.span()
```

高级regexp

| 表示联合多个regexp. A B两个regexp, A|B表示和A匹配或者跟B匹配。

^ 表示只匹配一行的开始行首, ^只有在开头才有此特殊意义。

\$ 表示只匹配一行的结尾

\A 表示只匹配第一行字符串的开头 ^匹配每一行的行首

\Z 表示只匹配行一行字符串的结尾 \$匹配第一行的行尾

\b 只匹配词的边界 例: \binfo\b 只会匹配"info" 不会匹配information

\B 表示匹配非单词边界

示例如下:

```
>>> print re.compile(r"\binfo\b").match("info ") #使用raw格式 \b表示单词
```

边界

```
<_sre.SRE_Match object at 0x817aa98>
```

```
>>> print re.compile("\binfo\b").match("info ") #没有使用raw \b表示退格符
```

号

```
None
```

```
>>> print re.compile("\binfo\b").match("\binfo\b ")
```

```
<_sre.SRE_Match object at 0x8174948>
```

分组(Group) 示例: re.compile("(a(b)c)d").match("abcd").groups() ('abc', 'b')

```
#!/usr/local/bin/python
```

```
import re
```

```
x = """
```

```
name: Charles
```

```
Address: BUPT
```

```
name: Ann
```

```
Address: BUPT
```

```
"""
```

```
#p = re.compile(r"^\name:(.*)\n^\Address:(.*)\n", re.M)
p = re.compile(r"^\name:(?P.*)\n^\Address:(?P.*)\n", re.M)
for m in p.finditer(x):
    print m.span()
    print "here is your friends list"
    print "%s, %s"%m.groups()
```

Compile Flag

用re.compile得到RegxObject时,可以有一些flag用来调整RegxObject的详细特征.

DOTALL, S 让.匹配任意字符,包括换行符\n

IGNORECASE, I 忽略大小写

LOCALES, L 让\w \W \b \B和当前的locale一致

MULTILINE, M 多行模式,只影响^和\$(参见上例)

VERBOSE, X verbose模式

阅读 (2409) | 评论 (0) | 转发 (0) |

上一篇: /proc/sysrq-trigger
下一篇: bash 字符串截取

0

相关热门文章

- | | |
|---------------------------------|---------------------------|
| A sample .exrc file for vi e... | linux dhcp peizhi roc |
| IBM System p5 服务器 HACMP ... | 关于Unix文件的软链接 |
| 游标的特征 | 求教这个命令什么意思,我是新... |
| busybox的httpd使用CGI脚本(Bu... | sed -e "/grep/d" 是什么意思... |
| Solaris PowerTOP 1.0 发布 | 谁能够帮我解决LINUX 2.6 10... |

给主人留下些什么吧! ^^

评论热议

登录后评论。
[登录](#) [注册](#)

- | | | | |
|--------------|-------------|---------------|-----------------|
| 1 python教程 | 4 昌平个人租房 | 7 入门基础教程 | 10 python功能 |
| 2 复式单身公寓 | 5 python模块 | 8 linux开发环境 | 11 python sys模块 |
| 3 python常用模块 | 6 python os | 9 python的os模块 | 12 python o: |



