

python os 模块

Python 的标准库中的 `os` 模块包含普遍的操作系统功能。如果你希望你的程序能够与平台无关的话，这个模块是尤为重要的。即它允许一个程序在编写后不需要任何改动，也不会发生任何问题，就可以在 Linux 和 Windows 下运行。

下面列出了一些在 `os` 模块中比较有用的部分。它们中的大多数都简单明了。

1. `os.sep` 可以取代操作系统特定的路径分割符。

2. `os.name` 字符串指示你正在使用的平台。比如对于 Windows，它是 `'nt'`，而对于 Linux/Unix 用户，它是 `'posix'`。

3. `os.getcwd()` 函数得到当前工作目录，即当前 Python 脚本工作的目录路径。

4. `os.getenv()` 和 `os.putenv()` 函数分别用来读取和设置环境变量。

5. `os.listdir()` 返回指定目录下的所有文件和目录名。

6. `os.remove()` 函数用来删除一个文件。

7. `os.system()` 函数用来运行 shell 命令。

8. `os.linesep` 字符串给出当前平台使用的行终止符。例如，Windows 使用 `'\r\n'`，Linux 使用 `'\n'` 而 Mac 使用 `'\r'`。

9. `os.path.split()` 函数返回一个路径的目录名和文件名。

10. `os.path.isfile()` 和 `os.path.isdir()` 函数分别检验给出的路径是一个文件还是目录。

11. `os.path.exists()` 函数用来检验给出的路径是否真地存在

os 和 os.path 模块

`os.listdir(dirname)`: 列出 `dirname` 下的目录和文件

`os.getcwd()`: 获得当前工作目录

`os.getcwd()`: 返回当前目录 ('.'))

`os.chdir(dirname)`: 改变工作目录到 `dirname`

`os.path.isdir(name)`: 判断 `name` 是不是一个目录, `name` 不是目录就返回 `false`

`os.path.isfile(name)`: 判断 `name` 是不是一个文件, 不存在 `name` 也返回 `false`

`os.path.exists(name)`: 判断是否存在文件或目录 `name`

`os.path.getsize(name)`: 获得文件大小, 如果 `name` 是目录返回 0L

`os.path.abspath(name)`: 获得绝对路径

`os.path.normpath(path)`: 规范 `path` 字符串形式

`os.path.split(name)`: 分割文件名与目录 (事实上, 如果你完全使用目录, 它也会将最后一个目录作为文件名而分离, 同时它不会判断文件或目录是否存在)

`os.path.splitext()`: 分离文件名与扩展名

`os.path.join(path, name)`: 连接目录与文件名或目录

`os.path.basename(path)`: 返回文件名

`os.path.dirname(path)`: 返回文件路径

python 中的字符串操作

#Python 字符串操作

'''1.复制字符串'''

#strcpy(sStr1,sStr2)

sStr1 = 'strcpy'

sStr2 = sStr1

sStr1 = 'strcpy2'

```

print sStr2

'''2.连接字符串'''

#strcat(sStr1,sStr2)
sStr1 = 'strcat'
sStr2 = 'append'
sStr1 += sStr2
print sStr1

'''3.查找字符'''

#strchr(sStr1,sStr2)
sStr1 = 'strchr'
sStr2 = 'r'
nPos = sStr1.index(sStr2)
print nPos

'''4.比较字符串'''

#strcmp(sStr1,sStr2)
sStr1 = 'strchr'
sStr2 = 'strch'
print cmp(sStr1,sStr2)

'''5.扫描字符串是否包含指定的字符'''

#strspn(sStr1,sStr2)
sStr1 = '12345678'
sStr2 = '456'
#sStr1 and chars both in sStr1 and sStr2
print len(sStr1 and sStr2)

'''6.字符串长度'''

#strlen(sStr1)
sStr1 = 'strlen'
print len(sStr1)

'''7.将字符串中的小写字符转换为大写字符'''

#strlwr(sStr1)
sStr1 = 'JCstrlwr'
sStr1 = sStr1.upper()
print sStr1

'''8.追加指定长度的字符串'''

#strncat(sStr1,sStr2,n)
sStr1 = '12345'
sStr2 = 'abcdef'

```

```

n = 3
sStr1 += sStr2[0:n]
print sStr1

'''9.字符串指定长度比较'''

#strncmp(sStr1,sStr2,n)
sStr1 = '12345'
sStr2 = '123bc'
n = 3
print cmp(sStr1[0:n],sStr2[0:n])

'''10.复制指定长度的字符'''

#strncpy(sStr1,sStr2,n)
sStr1 = ''
sStr2 = '12345'
n = 3
sStr1 = sStr2[0:n]
print sStr1

'''11.字符串比较，不区分大小写'''

#stricmp(sStr1,sStr2)
sStr1 = 'abcefg'
sStr2 = 'ABCEFG'
print cmp(sStr1.upper(),sStr2.upper())

'''12.将字符串前 n 个字符替换为指定的字符'''

#strnset(sStr1,ch,n)
sStr1 = '12345'
ch = 'r'
n = 3
sStr1 = n * ch + sStr1[3:]
print sStr1

'''13.扫描字符串'''

#strpbrk(sStr1,sStr2)
sStr1 = 'cekjgdklab'
sStr2 = 'gka'
nPos = -1
for c in sStr1:
    if c in sStr2:
        nPos = sStr1.index(c)
        break
print nPos

```

```
'''14.翻转字符串'''
```

```
#strrev(sStr1)
sStr1 = 'abcdefg'
sStr1 = sStr1[::-1]
print sStr1
```

```
'''15.查找字符串'''
```

```
#strstr(sStr1,sStr2)
sStr1 = 'abcdefg'
sStr2 = 'cde'
print sStr1.find(sStr2)
```

```
'''16.分割字符串'''
```

```
#strtok(sStr1,sStr2)
sStr1 = 'ab,cde, fgh,ijk'
sStr2 = ','
sStr1 = sStr1[sStr1.find(sStr2) + 1:]
```