OS

os.path.join()用法以及代码示例

python中的OS模块提供了与操作系统进行交互的功能，操作系统属于python的标准实用程序模块。该模块提供了使用依赖与操作系统的功能的便携式方法。os.path模块是python中os模块的sub-module，用于通用路径名操作

os.path.join()是python中方法，可以智能地连接一个或者多个路径组件。此方法将各个路径组成部分与每个非空部分之后的最后一个路径组成部分恰好用一个目录分隔符进行连接，如果要链接的最后一个路径组件为空，则将目录分隔符/放在末尾

如果路径组件表示绝对路径，那么将放弃所有先前连接的组件，并且从绝对路径组件进行连接。

```
用法：os.path.join(path,*paths)
```

参数：

path:代表文件系统路径的path-like对象 *paths:代表文件系统路径的path-like对象，他表示要链接的路径组件 path-like对象是表示路径的字符串或者字符对象

Note:python函数定义中的特殊语法*args(此处为*paths 用于将可变数量的参数传递给函数)

返回类型：此方法返回一个表示串联路径组件的字符串 代码：使用os.path.join()方法连接各种路径组件

```
# Python program to explain os.path.join() method

# importing os module
import os

# Path
path = "/home"

# Join various path components
print(os.path.join(path, "User/Desktop", "file.txt"))


# Path
path = "User/Documents"

# Join various path components
print(os.path.join(path, "/home", "file.txt"))

# In above example '/home'
# represents an absolute path
# so all previous components i.e User / Documents
# are thrown away and joining continues
# from the absolute path component i.e / home.
# Path
path = "/User"

# Join various path components
print(os.path.join(path, "Downloads", "file.txt", "/home"))

# In above example '/User' and '/home'
# both represents an absolute path
# but '/home' is the last value
# so all previous components before '/home'
# will be discarded and joining will
# continue from '/home'

# Path
path = "/home"

# Join various path components
print(os.path.join(path, "User/Public/", "Documents", ""))

# In above example the last
# path component is empty
# so a directory seperator ('/')
# will be put at the end
# along with the concatenated value
```

输出：

```
/home/User/Desktop/file.txt
/home/file.txt
/home
/home/User/Public/Documents/
```

OS模块常用的方法

import os

- 获得当前文件路径：os.getcwd()
- 新建一个文件夹：os.mkdir()
- 跳转当前文件路径：os.chdir()
- 获得路径下的所有文件名称：os.listdir(path)
- 返回是否是文件夹：os.path.isdir()
- 返回是否是文件：os.path.isfile()
- 将文件路径进行拆分：os.path.split()
- 修改文件名称：os.rename()