configpraser

* **. configpraser简介**

用于处理特定格式的文件，其本质还是利用open来操作文件。

配置文件的格式：

使用"[]"内包含section，section下为类似key-value的配置内容(例如：samba配置文件)

G:\Python项目实战\模块\configparser>start example.txt #基于windows平台

* 注释1

；注释2

[global] #节点

workgroup = WORKGROUP #值(key-value)

security = share

maxlog = 50

[public]

comment = stuff

public = True

pi = 3.1415926

* **. configpraser初始化**

使用时必须先初始化并读取配置文件

import configparser

config = configparser.ConfigParser()

config.read('example.txt',encoding='utf-8')

* **. configpraser常用方法**

1. 获取所有节点：

ret = config.sections() #读取配置文件里所有的"[]"信息

print(ret)



#输出:

['global', 'public']

1. 获取指定节点下的所有键值对：

ret = config.items('global')#获取指定节点的所有键值对

print(ret)



#输出：

[('workgroup', 'WORKGROUP'), ('security', 'share'), ('maxlog', '50')]

1. 获取指定节点下的所有键：

ret = config.options('public')#指定节点下的所有键

print(ret)

#输出：

['comment', 'public', 'pi']

1. 获取指定节点下指定key的值:

ret = config.get('global','workgroup')#获取指定节点下key的值

* ret = config.getint('global','maxlog')#获取指定节点下key值，必须为整数否则报错
* ret = config.getfloat('public','pi')#获取指定节点下key值，必须为浮点数否则报错
* ret = config.getboolean('public','public')#获取指定节点下key值，必须为布尔值否则报错

print(ret)

1. 检查，添加，删除节点

#检查

check = config.has\_section('global') #检查此节点下是否有值，返回布尔值

print(check)

#输出：

True

#添加节点

config.add\_section('local') #添加到内存

config.write(open('example.txt','w')) #写入文件中

ret = config.sections()

print(ret)

#输出：

['global', 'public', 'local']

#删除节点

config.remove\_section('local') #删除节点

config.write(open('example','w'))#重新写入文件

ret = config.sections()

print(ret)

#输出：

['global', 'public']

1. 检查，删除，设置指定组内的键值对

#检查

check = config.has\_option('public','comment')#检查节点下的某个键，返回布尔值

print(check)

输出:

True

#删除

config.remove\_option('global','workgroup')

config.write(open('example.txt','w'))

ret = config.options('global')

print(ret)

#输出：

['security', 'maxlog']

#设置指定节点内的键值对

ret1 = config.get('global','maxlog')

print(ret1)

config.set('global','maxlog','100')

config.write(open('example.txt','w'))

ret2 = config.get('global','maxlog')

print(ret2)

#输出：

50

100