

```
#####创建文件#####

# 来看一个好多软件的常见文档格式如下：

[DEFAULT]
ServerAliveInterval = 45
Compression = yes
CompressionLevel = 9
ForwardX11 = yes

[bitbucket.org]
User = hg

[topsecret.server.com]
Port = 50022
ForwardX11 = no
```

如果想用python生成一个这样的文档怎么做呢？

```
import configparser

config = configparser.ConfigParser()

config["DEFAULT"] = {'ServerAliveInterval': '45',
                     'Compression': 'yes',
                     'CompressionLevel': '9',
                     'ForwardX11': 'yes'
                     }

config['bitbucket.org'] = {'User': 'hg'}

config['topsecret.server.com'] = {'Host Port': '50022', 'ForwardX11': 'no'}

with open('example.ini', 'w') as configfile:

    config.write(configfile)
```

```
import configparser

config = configparser.ConfigParser()

#-----查找文件内容,基于字典的形式

print(config.sections())      #   []

config.read('example.ini')

print(config.sections())      #   ['bitbucket.org', 'topsecret.server.com']

print('bytebong.com' in config) # False
print('bitbucket.org' in config) # True
```

```
print(config['bitbucket.org']["user"])    # hg

print(config['DEFAULT']['Compression'])   #yes

print(config['topsecret.server.com']['ForwardX11'])    #no
```

```
print(config['bitbucket.org'])             #<Section: bitbucket.org>

for key in config['bitbucket.org']:        # 注意,有default会默认default的键
    print(key)

print(config.options('bitbucket.org'))     # 同for循环,找到'bitbucket.org'下所有键

print(config.items('bitbucket.org'))       #找到'bitbucket.org'下所有键值对

print(config.get('bitbucket.org','compression')) # yes      get方法Section下的key对应的value
```

```
import configparser

config = configparser.ConfigParser()

config.read('example.ini')

config.add_section('yuan')
```

```
config.remove_section('bitbucket.org')
config.remove_option('topsecret.server.com',"forwardx11")
```

```
config.set('topsecret.server.com','k1','11111')
config.set('yuan','k2','22222')
```

```
config.write(open('new2.ini', "w"))    #必须写,（文件一旦存在就放在那,只能放入新文件再覆盖）
```

例子

configparser

用途 用于生成和修改常见的配置文档

该模块适用于于配置文件的格式与 windows ini文件类似，可以包含一个或多个节（section），每个节可以有多个参数（键=值）

Import configparser  
config = configparser.ConfigParser()  
config.read(配置文件名)#只有先读才可以参看配置文件中内容（后面要修改也一样，改后必须写入文件）  
print(config.sections()) #获取所有的分支 默认没有打印出来

查找文件内容基于字典的形式

查找文件

增删改操作