



» [ConfigParserExamples](#)

» [ConfigParserExamples](#)

» [FrontPage](#)

» [RecentChanges](#)

» [FindPage](#)

» [HelpContents](#)

» [ConfigParserExamples](#)

Page

» Immutable Page

» [Info](#)

» [Attachments](#)

» More Actions:

User

» [Login](#)

Basic configparser usage

These are some examples on using [ConfigParser](#), assuming the following INI file...

```
[SectionOne]
Status: Single
Name: Derek
Value: Yes
Age: 30
Single: True

[SectionTwo]
FavoriteColor=Green
[SectionThree]
FamilyName: Johnson

[Others]
Route: 66
```

```
>>> import ConfigParser
>>> Config = ConfigParser.ConfigParser()
>>> Config
<ConfigParser.ConfigParser instance at 0x00BA9B20>
>>> Config.read("c:\\tomorrow.ini")
['c:\\tomorrow.ini']
>>> Config.sections()
['Others', 'SectionThree', 'SectionOne', 'SectionTwo']
>>>
```

Explanation: We first import the configparser, tell it to read the file, and get a listing of the sections. Sections are listed in square brackets [].

Next, we are going to get some settings, after defining a helper function.

The Function:

```
def ConfigSectionMap(section):
    dict1 = {}
    options = Config.options(section)
    for option in options:
        try:
            dict1[option] = Config.get(section, option)
            if dict1[option] == -1:
                DebugPrint("skip: %s" % option)
        except:
            print("exception on %s!" % option)
            dict1[option] = None
    return dict1
```

Now the code:

```
>>>Name = ConfigSectionMap("SectionOne")['name']
>>>Age = ConfigSectionMap("SectionOne")['age']
>>>print "Hello %s. You are %s years old." % (Name, Age)
Hello Derek. You are 30 years old.
```

This works great most of the time, but what about the "Value: Yes" and "Single: True" values? Those are booleans. They can be either True or False, Yes or No, 1 or 0, on or off. To read a boolean value, you use: `Config.getboolean(section, option)` Example, continuing from above:

```
>>>single = Config.getboolean("SectionOne", "single")
>>>single
```

True

You can also use `getint(section, option)` to get a number as an int. This may be easier to use than `int(Config.get(section, option))`. There is also `getfloat` which is used the same as `getint`, but, as you guessed, returns a float instead of an int.

Notes on reading an INI file

lines beginning with a semicolon ';' a pound sign '#' or the letters 'REM' (uppercase or lowercase) will be ignored. You may use these for comments if you want. You cannot put a comment on an option line. It will only be treated as a comment if it is at the beginning of the line!

Writing an INI file

When you write to an INI file, you will wipe out all comments.

Assuming the config file doesn't exist yet, this is the code to create one:

```
# lets create that config file for next time...
cfgfile = open("c:\\next.ini", 'w')

# add the settings to the structure of the file, and lets write it
out...
Config.add_section('Person')
Config.set('Person', 'HasEyes', True)
Config.set('Person', 'Age', 50)
Config.write(cfgfile)
cfgfile.close()
```

Advanced configparser usage

ExtendedInterpolation

Using `ExtendedInterpolation` one can make use of cross-chapter flexible parameter values. For instance, using the following ini file:

```
[SectionOne]
Param1: Hello
Param2: World

[SectionTwo]
Param1: ${SectionOne:Param1} ${SectionOne:Param2}
```

```
[SectionThree]
Alpha:    One
Bravo:    Two
Charlie:  ${Alpha} Mississippi
```

By setting `_interpolation` to `ExtendedInterpolation()` the values become dynamic.

```
>>> import configparser
>>> settings = configparser.ConfigParser()
>>> settings._interpolation = configparser.ExtendedInterpolation()
>>> settings.read('settings.ini')
['settings.ini']
>>> settings.sections()
['SectionOne', 'SectionTwo', 'SectionThree']
>>> settings.get('SectionTwo', 'Param1')
'Hello World'
>>> settings.get('SectionThree', 'Charlie')
'One Mississippi'
```

[CategoryDocumentation](#)

ConfigParserExamples (last edited 2014-05-12 12:01:02 by [Strijker](#))

- » [MoinMoin Powered](#)
- » [Python Powered](#)
- » [GPL licensed](#)
- » [Valid HTML 4.01](#)

[Unable to edit the page? See the FrontPage for instructions.](#)