



PYTHON

File Processing

Configparser - configuration file parser

FILE PROCESSING

Configparser - configuration file parser

FILE PROCESSING

- El módulo de la biblioteca estándar **configparser**
 - `import configparser`
- Permite generar archivos de configuración.
- Estructura:
 - Las secciones se identifican con nombres entre corchetes.
 - Las secciones contienen elementos que consisten en pares clave-valor.
 - Cada par se separa con dos puntos (:) o el signo igual (=).
 - Puede contener comentarios. Se escriben a continuación de un punto y coma (;) o una almohadilla (#).

La sección DEFAULT es común a todas las secciones

```
[DEFAULT]
host = localhost # This is a comment.
[mariadb]
name = hello
user = user
password = password
[redis]
port = 6379
db = 0
```

La estructura del archivo de configuración es muy similar a la de los archivos INI de Microsoft Windows.

FILE PROCESSING

■ Lectura:

- Clase **ConfigParser**:
- Método **read**
- Método **sections**
- Acceso:
 - **[sección][atributo]**

Se leen atributos
de DEFAULT

```
[DEFAULT]  
host = localhost # This is a comment.
```

```
[mariadb]  
name = hello  
user = user  
password = password
```

```
[redis]  
port = 6379  
db = 0
```

```
import configparser
```

```
config = configparser.ConfigParser()  
print(config.read('config.ini')) # ['config.ini']
```

```
print('Sections:', config.sections(), '\n') # Sections: ['mariadb', 'redis']
```

```
print('mariadb section:') # mariadb section:  
print('Host:', config['mariadb']['host']) # Host: localhost # This is a comment.  
print('Database:', config['mariadb']['name']) # Database: hello  
print('Username:', config['mariadb']['user']) # Username: user  
print('Password:', config['mariadb']['password'], '\n') # Password: password
```

```
print('redis section:') # redis section:  
print('Host:', config['redis']['host']) # Host: localhost # This is a comment.  
print('Port:', int(config['redis']['port'])) # Port: 6379  
print('Database number:', int(config['redis']['db'])) # Database number: 0
```

No aparece
DEFAULT

FILE PROCESSING

- Lectura de un diccionario:
 - Método **read_dict**
 - El API es el mismo que read.
- Más métodos:
 - Métodos **read_file** y **read_string**

```
import configparser

config = configparser.ConfigParser()

dict = {
    'DEFAULT': {
        'host': 'localhost'
    },
    'mariadb': {
        'name': 'hello',
        'user': 'root',
        'password': 'password'
    },
    'redis': {
        'port': 6379,
        'db': 0
    }
}

config.read_dict(dict)

print('Sections:', config.sections(), '\n')
```

FILE PROCESSING

- Creación:
 - Se debe tratar el objeto ConfigParser como un diccionario.
 - Cada sección es una clave.
 - Cada opción, un diccionario con los parámetros.
 - Método **write** para escribir el fichero

```
import configparser

config = configparser.ConfigParser()

config['DEFAULT'] = {'host': 'localhost'}
config['mariadb'] = {'name': 'hello',
                    'user': 'root',
                    'password': 'password'}
config['redis'] = {'port': 6379,
                  'db': 0}

with open('config.ini', 'w') as configfile:
    config.write(configfile)
```

FILE PROCESSING

- Admiten interpolación: utilizando una clave entre los caracteres **%** y **s**

```
[DEFAULT]
host = localhost

[mariadb]
name = hello
user = user
password = password

[redis]
port = 6379
db = 0
dsn = redis://%(host)s
```