01-Python intermediate

OS Module and methods with Examples

Posted on September 17, 2021
Last updated on February 12, 2023

1. Python OS Module

- Python OS module provides the facility to establish the `interaction between` the `user` and the `operating system`
- It provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

```
import os

# shows all attribute and method in this module.
print(dir(os))
```

1.1 Common OS functions

- `os.getcwd()`
  - Get current working directory(cwd)
- `os.listdir()`
  - list/display files and dir
- `os.chdir()`
  - Change dir/folder

```
print(os.getcwd())
# /home/user/Desktop

print(os.listdir())
# ['file1.py', 'dir1', 'file2.py]

# Change dir
os.chdir('dir1/')
```

- `os.mkdir()`
  - Create dir/folder, if exist shows error
- `os.makedirs()`
  - Create multi level dir

```
# Create dir
os.mkdir('newDir1')
os.makedirs('dir2/dir')
```

- `os.rename()`
  - Rename files n dir, if exist replace

```
os.rename('newDir1','newDir2')
```

- `os.remove()`
  - Delete only files
- `os.rmdir()`
  - Delete only empty dir
- `os.removedirs()`
  - Delete dir and its contents

```
os.remove('file1.py')

os.rmdir('newDir2')
os.rmdir('dir2/dir/')

os.removedirs('dir2/dir')
```

- `os.system()`
  - executing a shell command.

```
os.system('pwd')
os.system('ls')
```

1.2 OS Path

- `os.path.exists()`
  - True if path exist else False
- `os.path.getsize()`
  - Gives the size of the file/dir in bytes.
- `os.path.isdir()`
  - True if path is dir else False
- `os.path.isfile(path)`
  - True if path is file else False

```
os.path.exists('dir2')
size = os.path.getsize("file1.py")

os.path.isdir('dir2')
# True

os.path.isfile("C:\\Users\foo.csv")
# True
```

- `os.path.join()`
  - Joins one or more path components
- `os.path.abspath()`
  - Returns absolute/full path.

```
path = "/home"
print(os.path.join(path, "User/Desktop", "file.txt"))
# /home/User/Desktop/file.txt

print(os.getcwd()+'newfile.txt')    # '/' slash problem
print(os.path.join(os.getcwd(),'newfile'))  #better solution


print(os.path.abspath(__file__))
print(os.path.abspath('file.txt'))
# /home/User/Desktop/file.txt

print(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
print(os.path.dirname(os.path.dirname(os.path.abspath('file.txt'))))
# /home/User
```

1.3 Additional OS Fxn

- `os.environ`
  - Get all env(System,Users) variables
- `os.environ.get()`
  - Get the specific env

```
os.environ.get('HOME')
# '/home/uxer'

# Add a new environment variable
os.environ['myenv'] = 'envVar'
```

- `os.path.basename()`
  - base name of the specified path.
- `os.path.dirname()`
  - base dir of the specified path.
- `os.path.split()`
  - split dir and file name

```
print(os.path.basename('/temp/test.txt'))
# test.txt

print(os.path.dirname('/temp/test.txt'))
# /temp

print(os.path.split('tmp/temp/test.txt'))
# ('tmp/temp', 'test.txt')
```

- `os.walk()`
  - shows all dir, sub-dir, files from the given location

```
print(list(os.walk(os.getcwd())))
for dirpath, dirnames, filesname in os.walk('/home/dev/Desktop/'):
    print('curr path: ',dirpath)
    print('dirs: ',dirnames)
    print('files: ',filesname)
    print()
```

- `os.stat()`
  - This method is used to get status of the specified path.

```
# st_size= size of file
# st_mtime = last modified timestamp

print(os.stat('demo.json'))
print(os.stat('demo.json').st_size)

from datetime import datetime
# converting timestamp to redable form
print(datetime.fromtimestamp(os.stat('demo.json').st_mtime))
```

.

.

.

.