import os

print(os.name)

**os.getcwd():** Function os.getcwd(), returns the Current Working Directory(CWD) of the file used to execute the code, can vary from system to system.

import os

print(os.getcwd())

# To print absolute path on your system

# os.path.abspath('.')

# To print files and directories in the current directory

# on your system

# os.listdir('.')

**os.error:** All functions in this module raise OSError in the case of invalid or inaccessible file names and paths, or other arguments that have the correct type, but are not accepted by the operating system. os.error is an alias for built-in OSError exception.

import os

try:

# If the file does not exist,

# then it would throw an IOError

filename = 'GFG.txt'

f = open(filename, 'rU')

text = f.read()

f.close()

# Control jumps directly to here if

#any of the above lines throws IOError.

except IOError:

# print(os.error) will <class 'OSError'>

print('Problem reading: ' + filename)

# In any case, the code then continues with

# the line after the try/except

**os.rename():**A file old.txt can be renamed to new.txt, using the function os.rename(). The name of the file changes only if, the file exists and user has sufficient privilege permission to change the file.

import os

fd = "GFG.txt"

os.rename(fd,'New.txt')

os.rename(fd,'New.txt')

**os.path.basename(path) :** It is used to return the basename of the file . This function basically return the file name from the path given.

|  |
| --- |
| # basename function  importos  out =os.path.basename("/baz/foo")  print(out) |

**os.path.dirname(path) :** It is used to return the directory name from the path given. This function returns the name from the path except the path name.

# dirname function

import os

out = os.path.basename("/baz/foo")

print(out)

**os.path.isdir(path) :** This function specifies whether the path is existing directory or not.

# isdir function

import os

out = os.path.isdir("C:\\Users")

print(out)

**os.path.normcase(path) :** This function normalizes the case of the pathname specified. In Unix and Mac OS X system it returns the pathname as it is . But in Windows it converts the path to lowercase and forward slashes to backslashes.

# normcase function in windows

import os

out = os.path.normcase("/BAz")

print(out)

**os.name**

The os.name function gives the name of the OS module it imports. This differs based on the underlying Operating System.

# app.py

import os

print(os.name)

## ****os.environ****

The **environ** is not a function but a process parameter through which we can access environment variables of the system. Let’s see the following example.

import os

print(os.environ)

We can also print the HOME environment.

# app.py

import os

print(os.environ['HOME'])

## ****os.getcwd()****

The **getcwd**function of OS module will give us the current directory of the project.

# app.py

import os

print(os.getcwd())

If you want to make a new directory type the following code.

# app.py

import os

os.mkdir('newDir')