# OS Module in Python

The OS module in python provides functions for interacting with the operating system. OS, comes under Python’s standard utility modules. This module provides a portable way of using operating system dependent functionality. The \*os\* and \*os.path\* modules include many functions to interact with the file system.

Following are some functions in OS module:

**1. os.name:** This function gives the name of the operating system dependent module imported. The following names have currently been registered: ‘posix’, ‘nt’, ‘os2’, ‘ce’, ‘java’ and ‘riscos’

|  |
| --- |
| **import** os  print(os.name) |

# Python | os.path.join() method

**OS module** in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system dependent functionality. **os.path** module is sub-module of OS module in Python used for common pathname manipulation.

**os.path.join()** method in Python join one or more path components intelligently. This method concatenates various path components with exactly one directory separator (‘/’) following each non-empty part except the last path component. If the last path component to be joined is empty then a directory seperator (‘/’) is put at the end.  
If a path component represents an absolute path, then all previous components joined are discarded and joining continues from the absolute path component.

***Syntax:****os.path.join(path, \*paths)*

***Parameter:******path****: A path-like object representing a file system path.****\*path****: A path-like object representing a file system path. It represents the path components to be joined.  
A path-like object is either a string or bytes object representing a path.*

***Note:****The special syntax [\*args](https://www.geeksforgeeks.org/args-kwargs-python/" \t "https://www.geeksforgeeks.org/python-os-path-join-method/_blank) (here \*paths) in function definitions in python is used to pass a variable number of arguments to a function.*

***Return Type:****This method returns a string which represents the concatenated path components.*

|  |
| --- |
| # importing os module  **import** os    # Path  path **=** "/home"    # Join various path components  **print**(os.path.join(path, "User/Desktop", "file.txt"))      # Path  path **=** "User/Documents"    # Join various path components  **print**(os.path.join(path, "/home", "file.txt"))    # In above example '/home'  # represents an absolute path  # so all previous components i.e User / Documents  # are thrown away and joining continues  # from the absolute path component i.e / home.      # Path  path **=** "/User"    # Join various path components  **print**(os.path.join(path, "Downloads", "file.txt", "/home"))    # In above example '/User' and '/home'  # both represents an absolute path  # but '/home' is the last value  # so all previous components before '/home'  # will be discarded and joining will  # continue from '/home'    # Path  path **=** "/home"    # Join various path components  print(os.path.join(path, "User/Public/", "Documents", ""))    # In above example the last  # path component is empty  # so a directory seperator ('/')  # will be put at the end  # along with the concatenated value |

# Python | os.path.exists() method

**OS module** in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system dependent functionality. **os.path** module is sub module of **OS module** in python used for common path name manipulation.

**os.path.exists()** method in Python is used to check whether the specified path exists or not. This method can be also used to check whether the given path refers to an open file descriptor or not.

***Syntax:****os.path.exists(path)*

***Parameter:******path****: A path-like object representing a file system path. A path-like object is either a string or bytes object representing a path.*

***Return Type:****This method returns a Boolean value of class bool. This method returns True if path exists otherwise returns False.*

# importing os module

**import** os

# Specify path

path **=** '/usr/local/bin/'

# Check whether the specified

# path exists or not

isExist **=** os.path.exists(path)

print(isExist)

# Specify path

path **=** '/home/User/Desktop/file.txt'

# Check whether the specified

# path exists or not

isExist **=** os.path.exists(path)

print(isExist