Module Review Report

Zihua Li

Lecturer: Harikrishna Kuttivelil

CSE 20: Beginning Python

November 29, 2021

os.path Module

Module os.path is mainly used to obtain properties of files and paths.

It is very efficient when setting file path in different system environment and will be useful when creating a program which processes file names, properties, and paths. It has various kinds of method which offer lots of different kinds of function.

When comparing to pathlib module, which offers higher level path objects, os.path is easy to use and does not have too much complicated arguments and parameters to pass into. When the program is not doing higher level jobs, mainly focusing on basic path related manipulations, obtaining information, etc., it is not worth to use pathlib as using it will be a big talent wasted on a petty job, and will be pretty much time consuming. Using os.path will be extremely convenient when the program runs on different system environments as they have different styles of path representation, without using it, will be difficult to adapt the program in different system environments if accessing path related works.

Module Elements

os.path.abspath(path) Purpose: Returns absolute path

This takes in one argument, the relative path and returns its absolute path (will also work

if input is an absolute path but it will just return the same path you inputted, thus useless). This element is used in my demo program to get the absolute path from relative path inputted, as explained previously.

os.path.exists(path) Purpose: Returns True if path exists, otherwise return False

This takes in one argument, a relative path, or an absolute path, if exists return True, otherwise return False. This element is used in my demo program to check whether an inputted path exists or not, and also used everywhere to check does the path exists before running other functions, in case of resulting in error.

os.path.getsize(path) Purpose: Returns file size, if not exist returns an error

This takes in one argument, a relative path, or an absolute path, if not exists returns an error, if exists returns the size of the file in bytes. This element is used in my demo program to obtain the size of the file.

os.path.isabs(path) Purpose: Determines whether path is absolute or not

This takes in one argument, a relative path, or an absolute path, if path passed into is absolute, returns True, if path passed into is not absolute, returns False. This element is used in my demo program to check whether inputted path is absolute or not.

os.path.isfile(path) Purpose: Determines whether path is a file or not

This takes in one argument, a relative path, or an absolute path, if path passed into is a file, returns True, if path passed into is not a file, returns False. This element is used in my demo program to check whether inputted path is a file or not.

os.path.isdir(path) Purpose: Determines whether path is a directory or not

This takes in one argument, a relative path, or an absolute path, if path passed into is a

directory, returns True, if path passed into is not a directory, returns False. This element is used in my demo program to check whether inputted path is a file or not.

os.path.islink(path) Purpose: Determines whether path is a symbolic link or not

This takes in one argument, a relative path, or an absolute path, if path passed into is a symbolic link, returns True, if path passed into is not a symbolic link, returns False. This element is used in my demo program to check whether inputted path is a symbolic link or not.

Citation

os.path — Common pathname manipulations - [https://docs.python.org/3/library/os.path.html]

Instructions

1) Open the program in a command shell, the program will automatically run its main function.
2) The program will then show its main menu. There will be 7 options to select, select an option by entering its numeric order (“1)…” is option 1, if you want to select option one, enter 1).
3) If inputted option number is unacceptable (e.g. “328934”, “ABCD”, “$*#($”), the program will keep asking you to re-enter the option number.
4) The program will then ask you for the path, if path is not found (except for “2) Check whether path exists or not”), the program will keep asking you for the correct path.
5) The program will print out the expected results.
6) The program will then ask you if you want to go back to main menu or not. If you choose Yes, enter Y, the program will go back to main menu. If you choose No, enter N, the program will exit by itself.