Backslashes are used for paths in Windows, yet forward slashes appear to be used for paths in other operating system like Linux and OS X. (Chris, 2018)

Using forward slash character for file system was introduced by Unix around 1970. Popular operating systems like Linux and OS X are all based on Unix, so they followed the convention and used forward slash character as path character. The reason why Windows uses backslashes as path character is that the majority of the DOS tools of Windows were developed by IBM, and they already used the "/" character as a "switch" character, so they cannot use the same character to represent another meaning and have to choose a similar-looking character – "\" character – as the path character. (Chris, 2018)

When it comes to programming, the ability to handle paths on multiple systems is very important in practice. Since Windows has occupied the main part of operating system market and many developers use OS X to develop their programs, almost all these application developers expect their programs to be run on both Windows and other systems. Thus, they have to solve the problem of path inconsistency when they design their applications. For example, developers may need to fetch the paths of files or directories directly from the system and store it for other code to use. They need to make sure the target object is fetched successfully on different systems. Thus, handling paths on multiple systems really matters.

Python standard os library has many useful functions for listing and manipulating directory contents. The function listdir() gets the name list of all files and subdirectories according to the path passed. The function mkdir() creates a single directory at the input path. The functions unlink() and remove() delete a single file according to the path of the file. The function rmdir() deletes the empty directory at the path passed. All paths passed to the functions above must be strings representing the locations of target objects in the file system, and any error can interrupt the program. (Vuyisile, 2019)

Python pathlib module is a big improvement to path manipulation because pathlib is more straightforward to handle paths, pathlib is easier to create files and directories, and pathlib contains many helpful attribute functions. One key improvement of pathlib is that it contains all commonly used functions for path handling. In os module, you need to import os.path submodule if you want to join paths together or manipulate them. You need to import glob module if you want to get more specific file names. However, pathlib module can handle all the tasks mentioned or not mentioned above in just one import. This is extremely straightforward and easy to use. In addition, as Ahmed stated in his article, "Each Path object has multiple useful methods and attributes that perform operations previously handled by other libraries" (Ahmed, 2021). Python pathlib module has a number of useful attribute functions which are not contained by os module. These special functions can largely improve the designing of the program.

References

Ahmed, B. (2021, December 16). *Why You Should Start Using Pathlib as an Alternative to the OS Module*. https://towardsdatascience.com/why-you-should-start-using-pathlib-as-an-alternative-to-the-os-module-d9eccd994745

Chris, H. (2018, July 31). *Why Windows Uses Backslashes and Everything Else Uses Forward Slashes*. https://www.howtogeek.com/181774/why-windows-uses-backslashes-and-everything-else-uses-forward-slashes/

Vuyisile, N. (2019). *Working With Files in Python*. https://realpython.com/working-with-files-in-python/