Abstract

The Python module I want to change is the Pathlib module. The idea is that I think a copy function should be added to the Pathlib module. The new copy function will be designed to take two path object parameters, which are source and destination respectively. It is very normal for program developers to embed copy functionality into the code and adding this feature to Pathlib module also helps developers save the time to look for copy functionality from other uncommon modules. The new copy function takes two path objects as parameters and works with paths, which is related to the key goal of the Pathlib module.

Motivation

The current version and usage of the Pathlib module is insufficient since there is no copy functionality containing in the Pathlib module. The reason why this function should be added into the Pathlib module is that it is straightforward and easy for program developers to use when they design and implement their programs. For example, program developers may encounter the situation in which numerous similar files need to be copied from one folder to another before they are manipulated by the program so that program developers have the original files to compare. The most significant difference between this feature and the other is that developers only need to import one module to use the functionality without needing to waste time to search for other copy modules.

Rationale

The new functionality of copy will be added to the Pathlib module as a new public function which can be called by “pathlib.copy(source, destination)” code. Also, two required path objects will be passed to the function so that the source file can be copied to the destination directory. The name of the function is copy because there is no other functions or attributes called copy in the Pathlib module and the word “copy” can clearly represent the operation. In addition, copy operation needs at least two required paths to execute. One is the path of the file to be copied and the other is the path of the folder to store the copied file. Thus, two required parameters – source and destination – need to be passed to the copy function.

Specification

Program developers can use this new function by importing the Pathlib module. There are two ways to import the Pathlib module and use the Copy function. In the first method of importing, program developers import the Pathlib module the same as they import other modules – import pathlib. Then program developers can use “pathlib.copy(source, destination)” to call and execute

the function. Another way to import the copy function from the Pathlib module is “from pathlib import copy”. This style of importing simplifies the usage of the function. Program developers can just use “copy(source, destination)” to call and execute the function. There is no need to call the name of the Pathlib module. The two required parameters are path objects, which can be generated by calling the Path function of the Pathlib module. One example is showed as following:

```
import pathlib
source = pathlib.Path(‘test.txt’)
destination = pathlib.Path(‘test’)
pathlib.copy(source, destination)
```

The path of the file “test.txt” is assigned to the variable “source” by calling the Path function. The path of the directory “test” is assigned to the variable “destination” by calling the Path function. When the two variables are passed to the Copy function and the Copy function is executed, the file “test.txt” will be copied and move to the directory “test”. If the file “test.txt” does not exist in the path, the interpreter will report error. If the directory “test” already contains the file “test.txt” before the execution of the Copy function, the Copy function will still be executed and the original file “test.txt” will be covered by the new generated file “test.txt”.

Open issues

One potential weakness of this new design of adding the Copy function to the Pathlib module is that there are no other parameters to control and restrict the process of copying. Some parameter types may be specifying the encoding method to access the original file or specifying the number of copies to generate. In addition, the Copy function is not designed to copy a nonempty folder to another place. This is a crucial issue since if there are so many files in a folder need to be copied to another place, program developers may have to design certain structures to loop over the folder and copy the files other than directly copy the folder to the destination.