Section 11

Files

Mohammed Asir Shahid

2021-08-04

Contents

1 Filenames and Absolute/Relative File Paths

We'll learn about files, folders, and how Python can work with them.

If we want our data to persist after our program is finished, we need to save it to a file.

1.1 Fileames and File Paths

Each of our files has 2 key properties. The file name and the file path.

1.2 os module

The os module contains numerous file path related functions that we can use.

1.2.1 os.getcwd()

Every program has a setting called the current working directory, this tells us the directory that we are currently in. We can use the getcwd() function in order to find our current working directory.

```
import os

print(os.getcwd())
```

```
/home/mohammeds/Documents/Automate the Boring Stuff/Section 11: Files
```

1.2.2 os.chdir()

We can also change the current working directory using this chdir() function.

```
import os

print(os.getcwd())

os.chdir("/")

print(os.getcwd())
```

```
/home/mohammeds/Documents/Automate the Boring Stuff/Section 11: Files
/
```

1.3 Absolute and Relative Paths

There are two kinds of file paths, relative and absolute. The absolute file path begins with the root directory and gives you the complete location of the file. A relative file path is relative to the current working directory.

1.4 . and .. Folders

These are not real directories, but they can be used with relative paths. The single dot stands for this directory while two dots stands for the paret directory, the directory that your current working directory is in.

1.5 os.path module

1.5.1 os.path.join()

This is a join function inside of a path module inside of an os module. It takes several string arguments and returns a string value of a path for the os that you are using.

```
import os
print(os.path.join("folder1","folder2","folder3","file.png"))
```

```
folder1/folder2/folder3/file.png
```

1.5.2 os.path.abspath()

The abspath() function will return the absolute path of the path that you pass it.

```
import os

print(os.path.abspath("Section 11.org"))
```

```
/home/mohammeds/Documents/Automate the Boring Stuff/Section 11: Files/Section 11.org
```

1.5.3 os.path.isabs()

The isabs() function is a way to determine if a given path is absolute.

```
import os

print(os.path.abspath("Section 11.org"))

print(os.path.isabs("Section 11.org"))

print(os.path.abspath("/home/mohammeds/Documents/Automate the Boring Stuff/Section 11:

print(os.path.isabs("/home/mohammeds/Documents/Automate the Boring Stuff/Section 11: Fi
```

```
/home/mohammeds/Documents/Automate the Boring Stuff/Section 11: Files/Section 11.org
False
/home/mohammeds/Documents/Automate the Boring Stuff/Section 11: Files/Section 11.org
True
```

1.5.4 os.path.relpath()

This function will give you the relative path between two paths. The first argument is the file/directory we want to get to and the second is the directory we are already in.

```
import os

print(os.path.relpath("Section 11.org", "/usr/bin"))

print(os.path.relpath("Section 11.org", "/home/mohammeds/"))
```

```
../../home/mohammeds/Documents/Automate the Boring Stuff/Section 11: Files/Section 11.o
Documents/Automate the Boring Stuff/Section 11: Files/Section 11.org
```

1.5.5 os.path.dirname()

This function pulls out just the directory part of the path.

```
import os

print(os.path.dirname("/home/mohammeds/Documents/Automate the Boring Stuff/Section 11:
```

```
/home/mohammeds/Documents/Automate the Boring Stuff/Section 11: Files
```

1.5.6 os.path.basename()

This function pulls out just the last part of the path. Either the filename or just the directory itself if there is no file in the path.

```
import os

print(os.path.basename("/home/mohammeds/Documents/Automate the Boring Stuff/Section 11
```

```
Section 11.org
```

1.5.7 os.path.exists()

This function can check and see if the path you are passing actually exists.

```
import os

print(os.path.exists("/home/mohammeds/Documents/Automate the Boring Stuff/Section 11: F

print(os.path.exists("/home/mohammeds/Fake File Path"))
```

```
True
False
```

1.5.8 os.path.isfile()

We can use this function to see if what we pass to it is a file or not.

```
import os

print(os.path.isfile("/home/mohammeds/Documents/Automate the Boring Stuff/Section 11: F

print(os.path.isfile("/home/mohammeds/Fake File Path"))
```

```
True
False
```

1.5.9 os.path.isdir()

We can use this function to see if what we pass to it is a directory or not.

```
import os

print(os.path.isdir("/home/mohammeds/Documents/Automate the Boring Stuff/Section 11: Fi

print(os.path.isdir("/home/mohammeds/"))
```

```
False
True
```

1.5.10 os.path.getsize()

This function gives us the size of the file in bytes.

```
import os

print(os.path.getsize("/home/mohammeds/Documents/Automate the Boring Stuff/Section 11:
```

```
8057
```

1.6 os.listdir()

This function isn't in the path module. It can be passed a file path of a folder and then it'll return a list of strings of the filenames and folder names that it contains.

```
import os

print(os.listdir("/home/mohammeds/Documents"))
```

```
['[Chapman and Hall_CRC the R Ser] Albert, Jim_ Baumer, Benjamin S._ Marchi, Max - Anal
```

1.7 Example Code: Finding the total size of all files in a folder.

```
import os

totalSize=0

for file in os.listdir("/home/mohammeds/Documents"):
    if not os.path.isfile(os.path.join("/home/mohammeds/Documents", file)):
        continue
    totalSize+=os.path.getsize(os.path.join("/home/mohammeds/Documents", file))

print(totalSize)
```

```
382371482
```

1.8 os.makedirs()

This function can create directories for you.

```
import os

print(os.getcwd())

print(os.listdir())
```

```
os.makedirs("TestDir")

print(os.listdir())
```