Assignment

PW SKILLS

Q1. Which function is used to open a file? What are the different modes of opening a file? Explain each mode of file opening.

Answer.

The python open() function is used to open() internally stored files. It returns the contents of the file as python objects. Parameters: file_name.

The different mode of opening the file is as follows,

1. In 'r' mode, the file opens in the read mode. By default, if we don't specify the mode argument, it will be treated as read or 'r' mode. For Example, to open our file myfile.txt in 'r' mode:

open("myfile.txt") or **open ("myfile.txt", "r")**

2. In the 'w' mode, the file opens in write mode. It removes existing content, if present, in the file. If the specified file doesn't exist, it creates one with the specified name. For Example, to open our file myfile.txt in 'w' mode:

open("myfile.txt", "w")

3. In the 'a' mode, the file opens in append mode. It adds content to an existing file (or append at the end of the file). If the specified file doesn't exist, it creates one with the specified name. It does not remove the existing content from the file. For Example, to open our file myfile.txt in 'a' mode:

open("myfile.txt", "a")

4. In the 'r+' mode, the file opens in the read & write mode. It does not remove existing content, if present, in the file and does not create a file if not present in the directory. For Example, To open our file myfile.txt in 'r+'mode:

open("myfile.txt", "r+")

5. In the 'w+' mode, open the file in the read & write mode & remove existing content. If the file doesn't exist, it creates a new one. It does not remove existing content & if a file doesn't exist, it creates a new one. For Example, To open our file myfile.txt in 'w+' mode:

open("myfile.txt", "w+")

6. In the 'a+' mode, the file opens in the read & append mode. It does not remove existing content & If the file doesn't exist, it creates a new one. For Example, To open our file myfile.txt in a+ mode:

open("myfile.txt", "a+")

The modes discussed above are being used on a text file. To use these modes for a binary file, we need to use a different combination of file opening mode arguments. Using 'b' with any mode,

For example, ‘ab’, ‘rb’, ‘wb’, ‘rb+’, the file opens in binary mode. It is used for non-textual files like image, sound, executable (.exe) files.

Q2. Why close() function is used? Why is it important to close a file?

Answer:

The close() method closes an open file.

You should always close your files, in some cases, due to buffering, changes made to a file may not show until you close the file.

```
file.close()
```

Q3. Write a python program to create a text file. Write ‘I want to become a Data Scientist’ in that file. Then close the file. Open this file and read the content of the file.

Answer:

```
f= open("sample1.txt","w+")
f.write('I want to become a Data Scientist')
f.close()
```

Q4. Explain the following with python code: read(), readline() and readlines().

Answer:

The **read()** method returns the specified number of bytes from the file. Default is -1 which means the whole file.

Example:

```
f = open("sample.txt", "r")
print(f.read(33))
```

The **readline()** method returns one line from the file. You can also specify how many bytes from the line to return, by using the size parameter.

Example:

```
f = open("sample.txt", "r")
print(f.readline())
```

The **readlines()** method returns a list containing each line in the file as a list item. Use the hint parameter to limit the number of lines returned. If the total number of bytes returned exceeds the specified number, no more lines are returned.

Example:

```
f = open("sample.txt", "r")
print(f.readlines(33))
```

Q5. Explain why with statement is used with open(). What is the advantage of using with statement and open() together?

Answer: With statement is used to close the file automatically unlike the open() function where close() needs to be used at the end to close the opened file. you do not need to close the file at the end, because with would automatically close it for you.

Q6. Explain the write() and writelines() functions. Give a suitable example.

Answer: The **write()** method writes a specified text to the file.Where the specified text will be inserted depends on the file mode and stream position.
"a": The text will be inserted at the current file stream position, default at the end of the file.
"w": The file will be emptied before the text will be inserted at the current file stream position, default 0.
Example:

```
f = open("sample.txt", "a")
f.write("\nSee you soon!")
f.close()

#open and read the file after the appending:
f = open("sample.txt", "r")
print(f.read())
```

The writelines() method writes the items of a list to the file. Where the texts will be inserted depends on the file mode and stream position.

"a": The texts will be inserted at the current file stream position, default at the end of the file.
"w": The file will be emptied before the texts will be inserted at the current file stream position, default 0.

Example:

```
f = open("sample.txt", "a")
f.writelines(["\nSee you soon!", "\nOver and out."])
f.close()

#open and read the file after the appending:
f = open("sample.txt", "r")
print(f.read())
```