



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Experiment No. 5
Exploring Files and directories: Python program to append data to existing file and then display the entire file
Date of Performance:
Date of Submission:



Experiment No. 5

Title: Exploring Files and directories: Python program to append data to existing file and then display the entire file

Aim: To Exploring Files and directories: Python program to append data to existing file and then display the entire file

Objective: To Exploring Files and directories

Theory:

Directory also sometimes known as a folder are unit organizational structure in computer's file system for storing and locating files or more folders. Python now supports a number of APIs to list the directory contents. For instance, we can use the Path.iterdir, os.scandir, os.walk, Path.rglob, or os.listdir functions.

Python too supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files. The concept of file handling has stretched over various other languages, but the implementation is either complicated or lengthy, but alike other concepts of Python, this concept here is also easy and short. Python treats file differently as text or binary and this is important. Each line of code includes a sequence of characters and they form text file. Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma {,} or newline character. It ends the current line and tells the interpreter a new one has begun. Let's start with Reading and Writing files.

Working of open() function

We use open () function in Python to open a file in read or write mode. As explained above, open () will return a file object. To return a file object we use open() function along with two arguments, that accepts file name and the mode, whether to read or write. So, the syntax being: open(filename, mode). There are three kinds of mode, that Python provides and how files can be opened:

“ r “, for reading.



“ w “, for writing.

“ a “, for appending.

“ r+ “, for both reading and writing

Code:

```
import os
```

```
def create_directory(directory_name):
```

```
    try:
```

```
        os.makedirs(directory_name)
```

```
        print(f"Directory '{directory_name}' created successfully.")
```

```
    except FileExistsError:
```

```
        print(f"Directory '{directory_name}' already exists.")
```

```
def list_directory_contents(directory_name):
```

```
    try:
```

```
        contents = os.listdir(directory_name)
```

```
        print(f"Contents of directory '{directory_name}':")
```

```
        for item in contents:
```

```
            print(item)
```

```
    except FileNotFoundError:
```

```
        print(f"Directory '{directory_name}' not found.")
```

```
def remove_directory(directory_name):
```

```
    try:
```

```
        os.rmdir(directory_name)
```

```
        print(f"Directory '{directory_name}' removed successfully.")
```



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

```
except FileNotFoundError:
```

```
    print(f"Directory '{directory_name}' not found.")
```

```
except OSError as e:
```

```
    print(f"Error: {e}")
```

```
if __name__ == "__main__":
```

```
    directory_name = "test_directory"
```

```
    create_directory(directory_name)
```

```
    list_directory_contents(directory_name)
```

```
    remove_directory(directory_name)
```

```
IDLE Shell 3.10.11
File Edit Shell Debug Options Window Help
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023, 00:38:17) [MSC v
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informat
>>>
===== RESTART: C:/Users/Student/sbl5th.py =====
Directory 'test_directory' created successfully.
Contents of directory 'test_directory':
Directory 'test_directory' removed successfully.
>>>
```

```
def create_file(file_name):
```

```
    try:
```

```
        with open(file_name, 'x') as file:
```

```
            print(f"File '{file_name}' created successfully.")
```

```
    except FileExistsError:
```

```
        print(f"File '{file_name}' already exists.")
```

```
def write_to_file(file_name, content):
```



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

```
try:

    with open(file_name, 'w') as file:

        file.write(content)

    print(f"Content written to '{file_name}' successfully.")

except FileNotFoundError:

    print(f"File '{file_name}' not found.")

def read_from_file(file_name):

    try:

        with open(file_name, 'r') as file:

            content = file.read()

            print(f"Content of '{file_name}':")

            print(content)

    except FileNotFoundError:

        print(f"File '{file_name}' not found.")

def append_to_file(file_name, content):

    try:

        with open(file_name, 'a') as file:

            file.write(content)

            print(f"Content appended to '{file_name}' successfully.")

    except FileNotFoundError:

        print(f"File '{file_name}' not found.")

if __name__ == "__main__":

    file_name = "test_file.txt"
```



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

```
create_file(file_name)
```

```
write_to_file(file_name, "Hello, World!\n")
```

```
read_from_file(file_name)
```

```
append_to_file(file_name, "This is additional content.\n")
```

```
read_from_file(file_name)
```

```
File 'test_file.txt' created successfully.  
Content written to 'test_file.txt' successfully.  
Content of 'test_file.txt':  
Hello, World!  
  
Content appended to 'test_file.txt' successfully.  
Content of 'test_file.txt':  
Hello, World!  
This is additional content.
```

Conclusion: Directories and files have been explored.