جامـــــعـة المـــــسـتـقـبـل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قـســــــــم الانــظــمــة الــطـبـيـة الــذكـــيـــة

Lecture: (9)

(File Handling)

Subject: Object oriented programming I
Class: Second
Lecturer: Dr. Maytham N. Meqdad

File Handling

Python File Open

File handling is an important part of any web application.

Python has several functions for creating, reading, updating, and deleting files.

File Handling

The key function for working with files in Python is the `open()` function.

The `open()` function takes two parameters; *filename*, and *mode*.

There are four different methods (modes) for opening a file:

`"r"` - Read - Default value. Opens a file for reading, error if the file does not exist

`"a"` - Append - Opens a file for appending, creates the file if it does not exist

`"w"` - Write - Opens a file for writing, creates the file if it does not exist

`"x"` - Create - Creates the specified file, returns an error if the file exists

In addition you can specify if the file should be handled as binary or text mode

`"t"` - Text - Default value. Text mode

`"b"` - Binary - Binary mode (e.g. images)

Syntax

To open a file for reading it is enough to specify the name of the file:

```
f = open("demofile.txt")
```

The code above is the same as:

```
f = open("demofile.txt", "rt")
```

Because `"r"` for read, and `"t"` for text are the default values, you do not need to specify them.

Note: Make sure the file exists, or else you will get an error.

Open a File on the Server

Assume we have the following file, located in the same folder as Python:

```
demofile.txt

Hello! Welcome to demofile.txt
This file is for testing purposes.
Good Luck!
```

To open the file, use the built-in `open()` function.

The `open()` function returns a file object, which has a `read()` method for reading the content of the file:

Example

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

If the file is located in a different location, you will have to specify the file path, like this:

Example

Open a file on a different location:

```
f = open("D:\\myfiles\welcome.txt", "r")
print(f.read())
```

Read Only Parts of the File

By default the `read()` method returns the whole text, but you can also specify how many characters you want to return:

Example

Return the 5 first characters of the file:

```
f = open("demofile.txt", "r")
print(f.read(5))
```

Read Lines

You can return one line by using the `readline()` method:

Example

Read one line of the file:

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

By calling `readline()` two times, you can read the two first lines:

Example

Read two lines of the file:

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

By looping through the lines of the file, you can read the whole file, line by line:

Example

Loop through the file line by line:

```
f = open("demofile.txt", "r")
for x in f:
  print(x)
```

Close Files

It is a good practice to always close the file when you are done with it.

Example

Close the file when you are finish with it:

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

Note: 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.

Write to an Existing File

To write to an existing file, you must add a parameter to the `open()` function:

`"a"` - Append - will append to the end of the file

`"w"` - Write - will overwrite any existing content

Example

Open the file "demofile2.txt" and append content to the file:

```
f = open("demofile2.txt", "a")
f.write("Now the file has more content!")
f.close()

#open and read the file after the appending:
f = open("demofile2.txt", "r")
print(f.read())
```

Example

Open the file "demofile3.txt" and overwrite the content:

```
f = open("demofile3.txt", "w")
f.write("Woops! I have deleted the content!")
f.close()

#open and read the file after the overwriting:
f = open("demofile3.txt", "r")
print(f.read())
```

Note: the "w" method will overwrite the entire file.

Create a New File

To create a new file in Python, use the `open()` method, with one of the following parameters:

`"x"` - Create - will create a file, returns an error if the file exist

`"a"` - Append - will create a file if the specified file does not exist

`"w"` - Write - will create a file if the specified file does not exist

Example

Create a file called "myfile.txt":

```
f = open("myfile.txt", "x")
```

Result: a new empty file is created!

Example

Create a new file if it does not exist:

```
f = open("myfile.txt", "w")
```

Python Delete File

Delete a File

To delete a file, you must import the OS module, and run its `os.remove()` function:

Example

Remove the file "demofile.txt":

```
import os
os.remove("demofile.txt")
```

Check if File exist:

To avoid getting an error, you might want to check if the file exists before you try to delete it:

Example

Check if file exists, *then* delete it:

```
import os
if os.path.exists("demofile.txt"):
  os.remove("demofile.txt")
else:
  print("The file does not exist")
```

Delete Folder

To delete an entire folder, use the `os.rmdir()` method:

Example

Remove the folder "myfolder":

```
import os
os.rmdir("myfolder")
```

Note: You can only remove *empty* folders.

```
def add_student():
    name = input("Enter student name: ")
    grade = input("Enter student grade: ")

    with open('students.txt', 'a') as file:
        file.write(f"{name},{grade}\n")

    print(f"Student {name} added with grade {grade}.")

def view_students():
    try:
        with open('students.txt', 'r') as file:
            lines = file.readlines()
            if not lines:
                print("No students found.")
            else:
                print("Student List:")
                for line in lines:
                    name, grade = line.strip().split(',')
                    print(f"Name: {name}, Grade: {grade}")
    except FileNotFoundError:
        print("No students found.")

# Main loop
while True:
    print("\nLab System Menu:")
    print("1. Add Student")
    print("2. View Students")
    print("3. Exit")

    choice = input("Enter your choice (1/2/3): ")

    if choice == '1':
        add_student()
    elif choice == '2':
        view_students()
    elif choice == '3':
        print("Exiting program.")
        break
    else:
        print("Invalid choice. Please enter 1, 2, or 3.)
```

- This program provides a menu for adding students, viewing the list of students, and exiting the program. Student information is stored in a file named 'students.txt' in the format "name, grade". Feel free to modify the program to suit your specific requirements or let me know if you have any specific features in mind

```
import os

def create_file(filename):
    try:
        with open(filename, 'x'):
            print(f"File '{filename}' created successfully.")
    except FileExistsError:
        print(f"File '{filename}' already exists.")

def write_to_file(filename, content):
    with open(filename, 'w') as file:
        file.write(content)
    print(f"Content written to '{filename}' successfully.")

def read_file(filename):
    try:
        with open(filename, 'r') as file:
            content = file.read()
            print(f"Content of '{filename}':\n{content}")
    except FileNotFoundError:
        print(f"File '{filename}' not found.")

def delete_file(filename):
    try:
        os.remove(filename)
        print(f"File '{filename}' deleted successfully.")
    except FileNotFoundError:
        print(f"File '{filename}' not found.")
    except PermissionError:
        print(f"Permission denied. Unable to delete '{filename}'.")

# Main loop
while True:
    print("\nFile Handling Menu:")
    print("1. Create File")
    print("2. Write to File")
    print("3. Read File")
    print("4. Delete File")
    print("5. Exit")

    choice = input("Enter your choice (1/2/3/4/5): ")

    if choice == '1':
        filename = input("Enter the filename to create: ")
        create_file(filename)
    elif choice == '2':
        filename = input("Enter the filename to write to: ")
        content = input("Enter the content to write: ")
        write_to_file(filename, content)
    elif choice == '3':
        filename = input("Enter the filename to read: ")
        read_file(filename)
    elif choice == '4':
        filename = input("Enter the filename to delete: ")
        delete_file(filename)
    elif choice == '5':
        print("Exiting program.")
        break
    else:
        print("Invalid choice. Please enter 1, 2, 3, 4, or 5.")
```

- This program provides a simple menu to create, write to, read, and delete files. Make sure to run this program in a directory where you have the necessary permissions to create, write, and delete files. Feel free to customize it based on your specific needs.

```
def read_file(filename):
    try:
        with open(filename, 'r') as file:
            content = file.read()
            print(f"Content of '{filename}':\n{content}")
    except FileNotFoundError:
        print(f"File '{filename}' not found.")

def main():
    filename = input("Enter the filename to read: ")
    read_file(filename)

if __name__ == "__main__":
    main()
```

- This program prompts the user to enter a filename and then reads and prints the content of that file. It handles the case where the file is not found and uses a with statement to automatically close the file after reading.You can run this program, enter the filename when prompted, and it will display the content of the specified file. If the file is not found, it will print an appropriate message.
- ===

```
def write_to_file(filename, content):
    with open(filename, 'w') as file:
        file.write(content)
    print(f"Content written to '{filename}' successfully.")

def main():
    filename = input("Enter the filename to write to: ")
    content = input("Enter the content to write: ")
    write_to_file(filename, content)

if __name__ == "__main__":
    main()
```

- This program prompts the user to enter a filename and content. It then writes the provided content to the specified file. The with statement is used to ensure that the file is properly closed after writing.You can run this program, enter the filename and content when prompted, and it will write the content to the specified file. If the file already exists, it will be overwritten. If the file doesn't exist, a new file with the specified name will be created.
- ===

```
import os

def create_file(filename):
    try:
        with open(filename, 'x'):
            print(f"File '{filename}' created successfully.")
    except FileExistsError:
        print(f"File '{filename}' already exists.")

def delete_file(filename):
    try:
        os.remove(filename)
        print(f"File '{filename}' deleted successfully.")
    except FileNotFoundError:
        print(f"File '{filename}' not found.")
    except PermissionError:
        print(f"Permission denied. Unable to delete '{filename}'.")

def main():
    while True:
        print("\nFile Handling Menu:")
        print("1. Create File")
        print("2. Delete File")
        print("3. Exit")

        choice = input("Enter your choice (1/2/3): ")

        if choice == '1':
            filename = input("Enter the filename to create: ")
            create_file(filename)
        elif choice == '2':
            filename = input("Enter the filename to delete: ")
            delete_file(filename)
        elif choice == '3':
            print("Exiting program.")
            break
        else:
            print("Invalid choice. Please enter 1, 2, or 3.")

if __name__ == "__main__":
    main()
```

- This program provides a simple menu to create and delete files. You can run this program, choose options to create or delete files, and it will perform the respective operations. Note that the program assumes you have the necessary permissions to create and delete files in the specified directory.