**AI Day 12 Notes**

**Syed Mansoor ul Hassan Bukhari**

**Python File Handling:**

1. **Definition**: File handling in Python is used to read and write files. It allows us to create, read, update, and delete files.

# Open a file

file = open("demo.txt", "w")

file.write("Hello, world!")

file.close()

1. **Opening a File**: The open() function is used to open a file. The mode in which the file is opened is specified as a second argument.

file = open("demo.txt", "r") # Read mode

file = open("demo.txt", "w") # Write mode

file = open("demo.txt", "a") # Append mode

file = open("demo.txt", "b") # Binary mode

1. **Reading a File**: The read() method is used to read the content of a file.

file = open("demo.txt", "r")

content = file.read()

print(content)

file.close()

1. **Writing to a File**: The write() method is used to write content to a file.

file = open("demo.txt", "w")

file.write("Hello, world!")

file.close()

1. **Appending to a File**: The append() method is used to add content to the end of a file.

file = open("demo.txt", "a")

file.write("Appending text.")

file.close()

1. **File Modes**: Different modes can be used to open a file:
   * "r": Read mode (default)
   * "w": Write mode
   * "a": Append mode
   * "b": Binary mode
   * "x": Create mode (creates a new file, returns an error if the file exists)

file = open("demo.txt", "x")

1. **Reading Lines**: The readline() method reads a single line from the file.

file = open("demo.txt", "r")

line = file.readline()

print(line)

file.close()

1. **Reading All Lines**: The readlines() method reads all the lines from the file and returns them as a list.

file = open("demo.txt", "r")

lines = file.readlines()

for line in lines:

print(line)

file.close()

1. **Using with Statement**: The with statement is used to wrap the execution of a block of code. It ensures that the file is properly closed after its suite finishes.

with open("demo.txt", "r") as file:

content = file.read()

print(content)

1. **Checking if File Exists**: The os module can be used to check if a file exists.

import os

if os.path.exists("demo.txt"):

print("File exists")

else:

print("File does not exist")

1. **Deleting a File**: The os module can also be used to delete a file.

import os

if os.path.exists("demo.txt"):

os.remove("demo.txt")

else:

print("File does not exist")

1. **Creating a Directory**: The os module can be used to create a directory.

import os

os.mkdir("new\_directory")

1. **Removing a Directory**: The os module can also be used to remove a directory.

import os

os.rmdir("new\_directory")

1. **File Position**: The tell() method returns the current file position.

file = open("demo.txt", "r")

print(file.tell())

file.close()

1. **Changing File Position**: The seek() method changes the file position.

file = open("demo.txt", "r")

file.seek(0) # Move to the beginning of the file

print(file.read())

file.close()

1. **Truncating a File**: The truncate() method is used to resize the file to a specified size.

file = open("demo.txt", "w")

file.write("Hello, world!")

file.truncate(5)

file.close()

1. **File Attributes**: The os module can be used to get file attributes.

import os

file\_info = os.stat("demo.txt")

print(file\_info.st\_size) # Size of the file

print(file\_info.st\_mtime) # Last modification time