File Handling

Opening a File:

open(): Opens a file and returns a file object that can be used for reading, writing, or both.

Example:

```
file = open("filename.txt", "r")  # Open file for reading
```

Closing a File:

close(): Closes the file and releases the associated system resources. Example:

```
file.close()  # Close the file
```

Reading from a File:

read(): Reads the entire contents of a file as a string.
readline(): Reads a single line from the file.
readlines(): Reads all lines from the file and returns them as a list. Example:

```
content = file.read()  # Read the entire file content
line = file.readline()  # Read a single line
lines = file.readlines()  # Read all lines as a list
```

Writing to a File:

write(): Writes a string to the file.
writelines(): Writes a list of strings to the file. Example:

```
file.write("Hello, World!")  # Write a string to the file
file.writelines(["Line 1\n", "Line 2\n"])  # Write a list of strings
```

Appending to a File:

a mode: Opens the file in append mode, allowing you to add content at the end of the file.

Example:

```
file = open("filename.txt", "a")  # Open file in append mode
file.write("New line added!")  # Append content to the file
```

File Position:

seek(): Sets the file's current position to the given offset.

tell(): Returns the current position of the file. Example:

```
file.seek(0)  # Set the file position to the beginning
position = file.tell()  # Get the current file position
```

File Metadata:

name: Returns the name of the file.
mode: Returns the mode in which the file was opened. Example:

```
filename = file.name  # Get the name of the file
file_mode = file.mode  # Get the mode in which the file was opened
```

Checking File Existence:

```
import os
if os.path.exists("filename.txt"):
    print("File exists")
```

Renaming or Moving a File:

```
import os
os.rename("old_filename.txt", "new_filename.txt")
```

Deleting a File:

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

File Permissions:

```
import os
os.chmod("filename.txt", 0o755)  # Set file permissions to read, write, and execute for owner, and read and execute for others
```

Checking File Size:

```
import os
file_size = os.stat("filename.txt").st_size
```

Working with Directories:

```
import os
os.mkdir("new_directory")
os.rmdir("directory_to_remove")
```

Checking File or Directory:

```
import os
if os.path.isfile("filename.txt"):
    print("It is a file")
if os.path.isdir("directory"):
    print("It is a directory")
```

Changing Current Directory:

```
import os
os.chdir("/path/to/new_directory")
```

Listing Files and Directories:

```
import os
files = os.listdir("directory")
```

Walking a Directory Tree:

```
import os
for root, dirs, files in os.walk("directory"):
    for file in files:
        print(os.path.join(root, file))
```

File I/O Error Handling:

```
try:
    file = open("filename.txt", "r")
    # Perform file operations
except FileNotFoundError:
    print("File not found!")
except PermissionError:
    print("Permission denied!")
except IOError as e:
    print("IO error:", str(e))
finally:
```

```
    file.close()
```

File Object Attributes:

```
file = open("filename.txt", "r")
print(file.readable())  # Check if file is readable
print(file.writable())  # Check if file is writable
print(file.closed)  # Check if file is closed
```

File Object Methods:

```
file = open("filename.txt", "w")
print(file.seekable())  # Check if file object is seekable
file.write("Data")
file.flush()  # Flush the buffer to write the data immediately
file.truncate(10)  # Truncate the file to the specified size
```

Reading and Writing Binary Files:

```
file = open("filename.bin", "rb")
content = file.read()
file.close()

file = open("filename.bin", "wb")
file.write(content)
file.close()
```

File Iteration:

```
with open("filename.txt", "r") as file:
    for line in file:
        print(line)
```