File Handling in Python -

Introduction to File Handling

Definition:
File handling in Python allows you to read, write, and manipulate files directly from a Python program. It is essential for storing data persistently and interacting with external data sources.

Syntax:
No specific syntax, but the built-in `open()` function is used to work with files.

Example:
```
file = open("example.txt", "r")
print(file.read())
file.close()
```

Opening a File

Definition:
Files in Python are opened using the open() function. The function takes the filename and mode as arguments.

Syntax:
```
open("filename", "mode")
```

Example:
```
file = open("sample.txt", "w")
file.write("Hello World!")
file.close()
```

File Modes

Definition:
The mode parameter defines the operation to perform on the file:
- 'r' : Read (default)
- 'w' : Write
- 'a' : Append
- 'b' : Binary
- '+' : Read and Write

Syntax:
```
open("filename", "mode")
```

Example:
```
file = open("data.txt", "a")
file.write("\nAppending new data")
file.close()
```

Reading a File

Definition:
Python provides methods to read a file: read(), readline(), readlines().

Syntax:
file.read(size), file.readline(), file.readlines()

Example:
```
file = open("example.txt", "r")
print(file.read(10))  # Reads first 10 characters
file.close()
```

Writing to a File

Definition:
You can write content to a file using write() or writelines().

Syntax:
file.write(string), file.writelines(list)

Example:
```
file = open("example.txt", "w")
file.write("Python File Handling!")
file.close()
```

Appending to a File

Definition:
Appending allows you to add content to an existing file without overwriting it.

Syntax:
open("filename", "a")

Example:
```
file = open("example.txt", "a")
file.write("\nNew line added")
file.close()
```

Closing a File

Definition:
After working with a file, you should close it to free system resources.

Syntax:
file.close()

Example:
```
file = open("example.txt", "r")
print(file.read())
file.close()
```

Using with Statement

Definition:
Using 'with' ensures the file is closed automatically after use.

Syntax:
with open("filename", "mode") as file:

Example:
```
with open("example.txt", "r") as file:
    print(file.read())
```

File Object Methods

Definition:
Common methods: file.read(), file.write(), file.tell(), file.close().

Syntax:
file.method()

Example:
```
file = open("example.txt", "r")
print(file.tell())  # Current position
file.close()
```

Deleting a File

Definition:
You can delete a file using the os module.

Syntax:
os.remove('filename')

Example:
```
import os
os.remove("example.txt")
```