File Handling in Python

Pankaj Chouhan

www.codeswithpankaj.com

What is File Handling ?

File handling is an essential part of any programming language, allowing programs to read from and write to files stored on the system. Python provides built-in functions to create, read, write, and manipulate files efficiently.

Different Modes to Open a File

Python provides several modes to open a file :

Mode	Description
r	Read mode (default). Opens the file for reading. If the file doesn’t exist, it throws an error.
w	Write mode. Creates a new file if it doesn’t exist, or overwrites the existing file.
a	Append mode. Adds data to the end of an existing file.
r+	Read and write mode. The file must exist, otherwise, an error occurs.
w+	Write and read mode. Overwrites the file if it exists.
a+	Append and read mode. Adds new content while preserving existing content.
rb, wb, ab	Same as r, w, a, but for binary files.

Reading Files in Python

Methods

read()
Reads the entire file content.

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

readline()
Reads one line at a time.

```
file = open("example.txt", "r")
line = file.readline()
print(line)
file.close()
```

readlines()
Reads all lines and returns them as a list.

```
file = open("example.txt", "r")
lines = file.readlines()
print(lines)
file.close()
```

Pankaj Chouhan
www.codeswithpankaj.com

Writing to Files in Python

Methods

write()
Writes a string to a file

```
file = open("example.txt", "w")
file.write("Hello, World!\n")
file.close()
```

writelines()
Writes a list of strings to a file

```
file = open("example.txt", "w")
file.writelines(["Line 1\n", "Line 2\n"])
file.close()
```

Alternatively, use a with statement to automatically close the file:

```
with open("example.txt", "r") as file:
    print(file.read())  # No need to explicitly close the file
```

Working with File Pointers: seek() and tell()

tell()

Returns the current position of the file pointer.

```
file = open("example.txt", "r")
print(file.tell())  # Shows position
file.read(5)
print(file.tell())  # Updated position
file.close()
```

Using seek()

Moves the pointer to a specific position

```
file = open("example.txt", "r")
file.seek(5)  # Moves pointer to the 5th byte
print(file.read())
file.close()
```

Pickling & Unpickling (Storing Python Objects)

Pickling is a way to serialize (save) Python objects, and unpickling retrieves them.

Pickling (Saving Object)

```
import pickle

data = {"name": "Alice", "age": 25}
with open("data.pkl", "wb") as file:
    pickle.dump(data, file)
```

Unpickling (Loading Object)

```
with open("data.pkl", "rb") as file:
    data = pickle.load(file)
print(data)
```