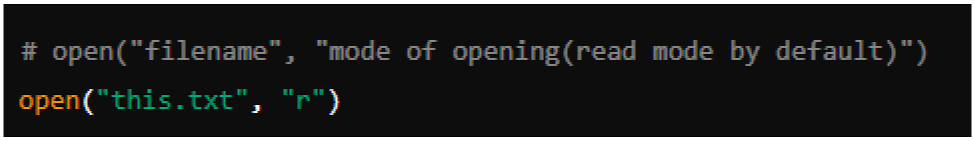
Chapter 9 – Opening and Reading Files in Python.

In Python, handling files is a common task. This chapter will help you understand how to open, read, and write files in an easy-to-understand way.

Opening a File

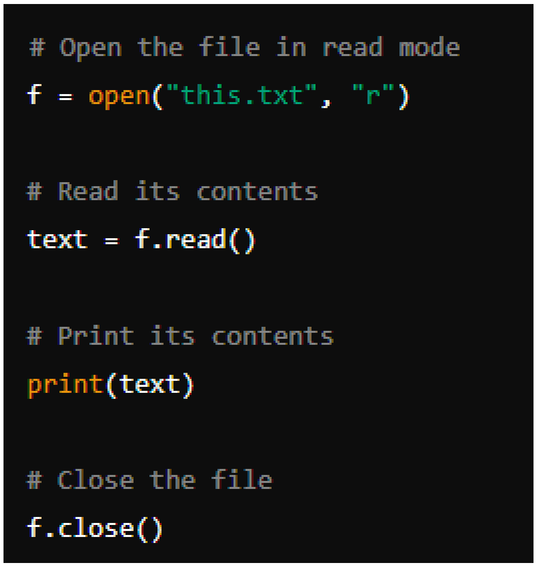
Python has an `open()` function for opening files. It takes two parameters: the filename and the mode of opening.



```
# open("filename", "mode of opening(read mode by default)")
open("this.txt", "r")
```

Reading a File in Python

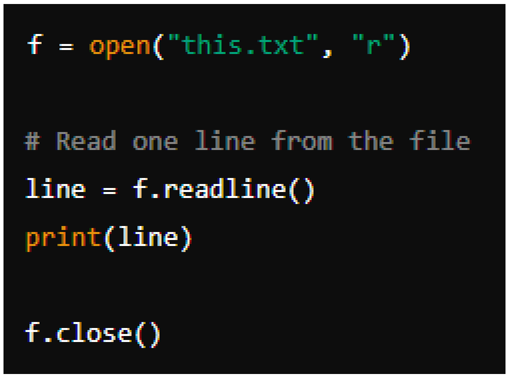
To read a file, you first need to open it in read mode, then read its contents and finally, close the file.



```
# Open the file in read mode
f = open("this.txt", "r")

# Read its contents
text = f.read()

# Print its contents
print(text)

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

Other Methods to Read the File

You can also use the `f.readline()` function to read one full line at a time.



```
f = open("this.txt", "r")

# Read one line from the file
line = f.readline()
print(line)

f.close()
```

Modes of Opening a File

Here are the different modes you can use to open a file:

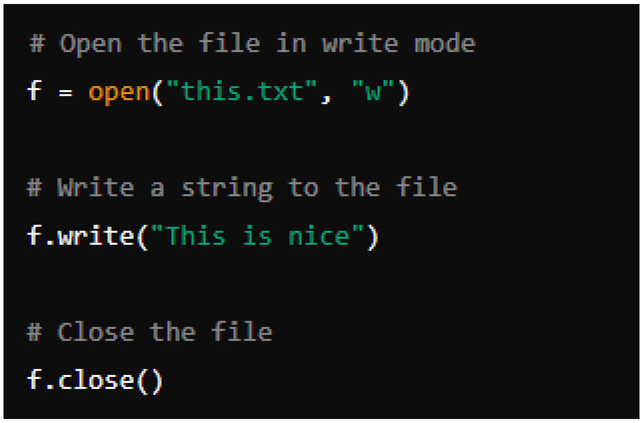
- `r` – open for reading (default mode)
- `w` – open for writing (creates a new file or truncates an existing file)
- `a` – open for appending (writes data at the end of the file)
- `+` – open for updating (reading and writing)

You can also open files in binary or text mode:

- `rb` – open for reading in binary mode
- `rt` – open for reading in text mode

Writing to Files in Python

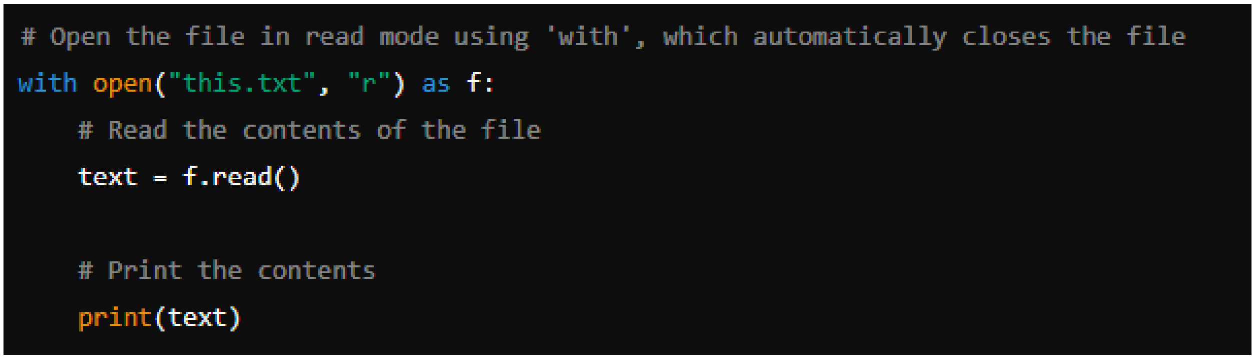
To write to a file, you first open it in write or append mode. Then, use the `f.write()` method to write to the file.



```
# Open the file in write mode
f = open("this.txt", "w")

# Write a string to the file
f.write("This is nice")

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

Using the `with` Statement

The best way to open and close a file automatically is to use the `with` statement. This ensures the file is properly closed after its suite finishes.



```python
# Open the file in read mode using 'with', which automatically closes the file
with open("this.txt", "r") as f:
    # Read the contents of the file
    text = f.read()

    # Print the contents
    print(text)
```

The `with` statement is beneficial because it makes your code cleaner and handles the file closing automatically, even if an error occurs during file operations.