File handling in Python involves interacting with files on a computer's file system, allowing for operations such as creating, reading, writing, appending, and deleting files. This is crucial for data persistence and managing external data sources.

Core Concepts:

* **Opening a File:** The open() function is used to open a file, returning a file object. It requires the file path and the mode in which the file should be opened.

Python

file = open("filename.txt", "mode")

* **Modes:**
  + "r": Read mode (default). Error if file doesn't exist.
  + "w": Write mode. Creates file if it doesn't exist, overwrites if it does.
  + "a": Append mode. Creates file if it doesn't exist, adds to end if it does.
  + "x": Create mode. Creates file, errors if it already exists.
  + "b": Binary mode (e.g., "rb", "wb").
  + "t": Text mode (default, e.g., "rt", "wt").
* **Working with Files (Read/Write):**

Once opened, the file object provides methods for interacting with its content.

* + **Reading:**
    - read(): Reads the entire file content.
    - readline(): Reads a single line.
    - readlines(): Reads all lines into a list.
  + **Writing:**
    - write(string): Writes a string to the file.
    - writelines(list\_of\_strings): Writes a list of strings (lines) to the file.
* **Closing a File:**

It is essential to close files after operations to release system resources.

Python

file.close()

* with Statement (Recommended): The with statement ensures that the file is automatically closed, even if errors occur.

Python

with open("filename.txt", "r") as file:  
 content = file.read()  
 # File is automatically closed here