**File Handling**

File handling in Python involves using built-in functions to create, read, update, and delete files. Python provides a set of functions to handle files in various modes, such as reading from a file, writing to a file, appending to a file, and so on. Here's a comprehensive guide to file handling in Python:

**Opening a File**

To open a file, use the open() function. This function returns a file object and is used with two arguments: the file name and the mode.

file = open('example.txt', 'r') # Opens the file in read mode

**Modes for Opening Files**

* 'r': Read (default mode). Opens the file for reading, and the file must exist.
* 'w': Write. Opens the file for writing, creating a new file or truncating the existing file to zero length.
* 'a': Append. Opens the file for writing, creating a new file if it does not exist, and appends data to the end of the file.
* 'b': Binary mode. Used with other modes (e.g., 'rb', 'wb') for binary files.
* 't': Text mode (default mode). Used with other modes (e.g., 'rt', 'wt').

**Reading from a File**

You can read the content of a file using various methods like read(), readline(), and readlines().

#Open a File on the Server

Assume we have the following file, located in the same folder as Python:

demofile.txt

Hello! Welcome to demofile.txt

This file is for testing purposes.

Good Luck!

To open the file, use the built-in open() function.

The open() function returns a file object, which has a read() method for reading the content of the file:

Example

f = open("demofile.txt", "r")

print(f.read())

If the file is located in a different location, you will have to specify the file path, like this:

Example

Open a file on a different location:

f = open("D:\\myfiles\welcome.txt", "r")

print(f.read())

**Read Only Parts of the File**

By default the read() method returns the whole text, but you can also specify how many characters you want to return:

Example

Return the 5 first characters of the file:

f = open("demofile.txt", "r")

print(f.read(5))

**Read Lines**

You can return one line by using the readline() method:

Read one line of the file:

f = open("demofile.txt", "r")

print(f.readline())

By calling readline() two times, you can read the two first lines:

Example

Read two lines of the file:

f = open("demofile.txt", "r")

print(f.readline())

print(f.readline())

By looping through the lines of the file, you can read the whole file, line by line:

Example

Loop through the file line by line:

f = open("demofile.txt", "r")

for x in f:

print(x)

**Close Files**

It is a good practice to always close the file when you are done with it.

Example

Close the file when you are finish with it:

f = open("demofile.txt", "r")

print(f.readline())

f.close()

**Writing to a File**

To write to an existing file, you must add a parameter to the open() function:

* "a" - Append - will append to the end of the file
* "w" - Write - will overwrite any existing content

Example

Open the file "demofile2.txt" and append content to the file:

f = open("demofile2.txt", "a")

f.write("Now the file has more content!")

f.close()

#open and read the file after the appending:

f = open("demofile2.txt", "r")

print(f.read())

Example

Open the file "demofile3.txt" and overwrite the content:

f = open("demofile3.txt", "w")

f.write("Woops! I have deleted the content!")

f.close()

#open and read the file after the overwriting:

f = open("demofile3.txt", "r")

print(f.read())

Note: the "w" method will overwrite the entire file.

**Create a New File**

To create a new file in Python, use the open() method, with one of the following parameters:

* "x" - Create - will create a file, returns an error if the file exist
* "a" - Append - will create a file if the specified file does not exist
* "w" - Write - will create a file if the specified file does not exist

Example

Create a file called "myfile.txt":

f = open("myfile.txt", "x")

Result: a new empty file is created!

Example

Create a new file if it does not exist:

f = open("myfile.txt", "w")

**Delete a File**

To delete a file, you must import the OS module, and run its os.remove() function:

Example

Remove the file "demofile.txt":

import os

os.remove("demofile.txt")

Check if File exist:

To avoid getting an error, you might want to check if the file exists before you try to delete it:

Example

Check if file exists, then delete it:

import os

if os.path.exists("demofile.txt"):

os.remove("demofile.txt")

else:

print("The file does not exist")

**Delete Folder**

To delete an entire folder, use the os.rmdir() method:

Example

Remove the folder "myfolder":

import os

os.rmdir("myfolder")

Note: You can only remove empty folders.