Python - Write to File

Writing to a file involves opening the file in a specific mode, writing data to it, and then closing the file to ensure that all data is saved and resources are released. Python provides a built-in function open() to handle file operations and various methods for writing data.

Opening a File for Writing

Opening a file for writing is the first step in performing write operations in Python. The open() function is used to open files in different modes, each suited for specific use cases.

The open() Function

The open() function in Python is used to open a file. It requires at least one argument, the name of the file, and can take an optional second argument that specifies the mode in which the file should be opened.

File Modes for Writing

Following are the main modes you can use to open a file for writing −

- **w (Write mode) −** Opens the file for writing. If the file exists, it truncates (empties) the file before writing. If the file does not exist, it creates a new file.
- **a (Append Mode) −** Data is written at the end of the file. If the file does not exist, it creates a new file.
- **x (Exclusive Creation Mode) −** Opens the file for exclusive creation. If the file already exists, the operation fails.
- **b (Binary Mode) −** When used with other modes, opens the file in binary mode.
- **+ (Update Mode) −** Opens the file for updating (reading and writing).

Example: Opening a File in Write Mode

This mode is used when you want to write data to a file, starting from scratch each time the file is opened −

</> Open Compiler

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

```
file.close()
print ("File opened successfully!!")
```

Following is the output obtained −

```
File opened successfully!!
```

Example: Opening a File in Append Mode

This mode is used when you want to add data to the end of the file without altering its existing contents −

</> Open Compiler

```
file = open("example.txt", "a")
file.write("Appending this line.\n")
file.close()
print ("File opened successfully!!")
```

This will produce the following result −

```
File opened successfully!!
```

Writing to a File Using write() Method

The write() method is used to write a single string to a file. This makes it suitable for various text-based file operations.

The write() method takes a single argument: the string that you want to write to the file. It writes the exact content of the string to the file without adding any additional characters, such as newlines.

Example

In the following example, we are opening the file "example.txt" in write mode. We then use the write() method to write a string to the file −

</> Open Compiler

```
# Open a file in write mode
with open("example.txt", "w") as file:
   file.write("Hello, World!\n")
   file.write("This is a new line.\n")
print ("File opened successfully!!")
```

Following is the output of the above code −

```
File opened successfully!!
```

Learn **Python** in-depth with real-world projects through our **Python certification course**. Enroll and become a certified expert to boost your career.

Writing to a File Using writelines() Method

The writelines() method is used to write a list of strings to a file. Each string in the list is written to the file sequentially without adding any newline characters automatically.

Example

In this example, we are creating a list of strings, lines, with each string ending in a newline character. We then open a file "example.txt" in write mode and use the writelines() method to write all the strings in the list to the file in one operation −

</> Open Compiler

```
# List of lines to write to the file
lines = ["First line\n", "Second line\n", "Third line\n"]

# Open a file in write mode
with open("example.txt", "w") as file:
    file.writelines(lines)

print ("File opened successfully!!")
```

The output obtained is as shown below −

```
File opened successfully!!
```

Writing to a New File

Writing to a new file in Python involves creating a new file (or overwriting an existing one) and writing the desired content to it. Here, we will explain the steps involved in writing to a new file −

- **Open the File −** Use the open() function to create or open a file in write mode ("w" or "wb" for binary files).
- **Write Data −** Use the write() or writelines() method to write data to the file.
- **Close the File −** Ensure the file is properly closed after writing, generally using the "with" statement for automatic handling.

Example

In the example below, we create a "foo.txt" file and write given content in that file and finally close that file −

```
# Open a file
fo = open("foo.txt", "w")
fo.write( "Python is a great language.\nYeah its great!!\n")

# Close opened file
fo.close()
```

If you open this file with any text editor application such as Notepad, it will have the following content −

```
Python is a great language.
Yeah its great!!
```

Writing to a New File in Binary Mode

By default, read/write operations on a file object are performed on text string data. If we need to handle files of different types, such as media files (mp3), executables (exe), or pictures (jpg), we must open the file in binary mode by adding the 'b' prefix to the read/write mode.

Writing Binary Data to a File

To write binary data to a file, open the file in binary write mode ('wb'). The following example demonstrates this −

```
# Open a file in binary write mode
with open('test.bin', 'wb') as f:
   # Binary data
   data = b"Hello World"
   f.write(data)
```

Converting Text Strings to Bytes

Conversion of a text string to bytes can be done using the encode() function. This is useful when you need to write text data as binary data −

```
# Open a file in binary write mode
with open('test.bin', 'wb') as f:
   # Convert text string to bytes
   data = "Hello World".encode('utf-8')
   f.write(data)
```

Writing to an Existing File

When an existing file is opened in write mode ('w'), its previous contents are erased. Opening a file with write permission treats it as a new file. To add data to an existing file without erasing its contents, you should open the file in append mode ('a').

Example

The following example demonstrates how to open a file in append mode and add new text to it −

```
# Open a file in append mode
fo = open("foo.txt", "a")
text = "TutorialsPoint has a fabulous Python tutorial"
fo.write(text)

# Close opened file
fo.close()
```

If you open this file with any text editor application such as Notepad, it will have the following content −

```
Python is a great language.
Yeah its great!!
```

TutorialsPoint has a fabulous Python tutorial

Writing to a File in Reading and Writing Modes

When a file is opened for writing using 'w' or 'a', it is not possible to perform write operations at any earlier byte position in the file. The 'w+' mode, however, allows both reading and writing operations without closing the file. The seek() function is used to move the read/write pointer to any desired byte position within the file.

Using the seek() Method

The seek() method is used to set the position of the read/write pointer within the file. The syntax for the seek() method is as follows −

```
fileObject.seek(offset[, whence])
```

Where,

- **offset** − This is the position of the read/write pointer within the file.
- **whence** − This is optional and defaults to 0 which means absolute file positioning, other values are 1 which means seek relative to the current position and 2 means seek relative to the file's end.

Example

The following program demonstrates how to open a file in read-write mode ('w+'), write some data, seek a specific position, and then overwrite part of the file's content −

```
# Open a file in read-write mode
fo = open("foo.txt", "w+")

# Write initial data to the file
fo.write("This is a rat race")

# Move the read/write pointer to the 10th byte
fo.seek(10, 0)

# Read 3 bytes from the current position
data = fo.read(3)

# Move the read/write pointer back to the 10th byte
```

```
fo.seek(10, 0)

# Overwrite the existing content with new text
fo.write('cat')

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

If we open the file in read mode (or seek to the starting position while in 'w+' mode) and read the contents, it will show the following −

```
This is a cat race
```