Python File Methods

Python has a set of methods available for the file object.

|  |  |
| --- | --- |
| **Method** | **Description** |
| **Close()** | Closes the file. You should always close your files, in some cases, due to buffering, changes made to a file may not show until you close the file.  **Syntax: file.close()** |
| **Detach()** | Returns the separated raw stream from the buffer.  **Syntax: file.detach()** |
| **Fileno()** | Returns the file descriptor of the stream, as a number.  An error will occur if the operator system does not use a file descriptor.  **Syntax: file.fileno()** |
| **Flush()** | Flushes the internal buffer.  **Syntax: file.flush()**  Ej:  f = open("myfile.txt", "a") f.write("Now the file has one more line!") f.flush() f.write("...and another one!") |
| **Isatty()** | Returns whether the file stream is interactive or not.  Returns True if the file stream is interactive, for example: connected to a terminal device.  **Syntax: file.isatty()** |
| **Read()** | Returns the file content.  returns the specified number of bytes from the file. Default is -1 which means the whole file.  **Syntax: file.read()** |
| **Readable()** | Returns whether the file stream can be read or not.  returns True if the file is readable, False if not.  **Syntax: file.readable()** |
| **Readline()** | Returns one line from the file.  ou can also specified how many bytes from the line to return, by using the size parameter.  **Syntax: file.readline(size)**  Ej: f = open("demofile.txt", "r") print(f.readline(5)) |
| **Readlines()** | Returns a list of lines from the file.  Use the hint parameter to limit the number of lines returned. If the total number of bytes returned exceeds the specified number, no more lines are returned.  **Syntax: file.readlines(hint) or file.readlines()** |
| **Seek()** | Change the file position.  The seek() method sets the current file position in a file stream.  The seek() method also returns the new postion.  **Syntax: *file*.seek(*offset*)**  Offset: Required. A number representing the position to set the current file stream position.  Ej:  f = open("demofile.txt", "r") print(f.seek(4)) |
| **Seekable()** | Returns whether the file allows us to change the file position.  The seekable() method returns True if the file is seekable, False if not.  A file is seekable if it allows access to the file stream, like the [seek()](https://www.w3schools.com/python/ref_file_seek.asp) method.  **Syntax: file.seekable()** |
| **Tell()** | Returns the current file position.  The tell() method returns the current file position in a file stream.  **Syntax:file.tell()** |
| **Truncate()** | Resizes the file to a specified size.  Resizes the file to the given number of bytes.  If the size is not specified, the current position will be used.  **Syntax: file.truncate(size)** |
| **Writable()** | Returns whether the file can be written to or not.  Returns True if the file is writable, False if not.  A file is writable if it is opened using "a" for append or "w" for write.  **Syntax: file.writable()** |
| **Write()** | Writes the specified string to the file.  Where the specified text will be inserted depends on the file mode and stream position.  "a": The text will be inserted at the current file stream position, default at the end of the file.  "w": The file will be emptied before the text will be inserted at the current file stream position, default 0.  **Syntax: file.write(byte)**  Byte=The text or byte object that will be inserted. |
| **Writelines()** | Writes a list of strings to the file.  **Syntax: file.writelines(list)**  List=The list of texts or byte objects that will be inserted.  Ej:  f = open("demofile3.txt", "a") f.writelines(["\nSee you soon!", "\nOver and out."]) f.close() |