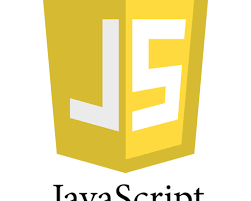
**Manejo de archivos en JavaScript**

.

JavaScript es un lenguaje de programación popular que le permite manejar archivos en el navegador. ¡Aprendamos cómo!

El entorno NodeJS se utiliza para diferentes scripts que incluyen el manejo de archivos. NodeJS no es más que un entorno para ejecutar código JavaScript. Espero que tengas un básico [comprensión de NodeJS](https://geekflare.com/es/mongodb-and-nodejs/).

Pasemos al tutorial para aprender sobre el manejo de archivos en JavaScript.

## Manejo de archivos en JavaScript

El manejo de archivos incluye diferentes operaciones como **la creación**, **lectura**, **actualización**, **renombrando**y **eliminando**. Tenemos que acceder a los archivos del sistema, lo que no nos es posible escribir desde cero. Entonces, NodeJS proporciona un módulo llamado **fs (sistema de archivos)**para el manejo de archivos.

Veamos diferentes métodos del **fs**módulo.

## **fs.open()**

El método fs.open() tomará dos argumentos **camino**y  **modo**.

El **camino** se utiliza para localizar el archivo.

El argumento **modo** se utiliza para abrir el archivo en diferentes modos como **adjuntar, escribir,** y **leyendo.**

Si abre cualquier archivo en un modo específico, entonces puede realizar solo un tipo de operación correspondiente al modo que ha pasado al método. Veamos la lista de modos y operaciones correspondientes.

|  |  |
| --- | --- |
| **Moda** | **Operación** |
| 'r' | Abre un archivo en modo lectura |
| 'un' | Abre un archivo en modo adjunto |
| 'w' | Abre un archivo en modo de escritura |
| 'a +' | Abre un archivo en modo de adición y lectura |
| 'w +' | Abre un archivo en modo de escritura y lectura |
| 'r +' | Abre un archivo en modo lectura y escritura |

Si el archivo no existe en la ruta dada, creará un nuevo archivo vacío. Veamos el código para abrir un archivo en diferentes modos.

const fs = require("fs");

fs.open("sample.txt", "w", (err, file) => {

if (err) throw err;

console.log(file);

});

El método fs.open() arrojará un error si el archivo no existe al abrir en **lectura**modo. Creará un nuevo archivo vacío en **la escritura**y  **adjuntando**Modos.

Podemos realizar diferentes operaciones sobre el archivo abierto. Escribiremos un programa completo al final de este tutorial después de aprender algunos métodos más esenciales del **fs**módulo.

## **fs.appendFile()**

El método fs.appendFile() se utiliza para agregar el contenido al final del archivo. Si el archivo no existe en la ruta dada, creará uno nuevo. Agregue algo de contenido al archivo usando el siguiente código.

const fs = require("fs");

fs.appendFile("sample.txt", "Appending content", (err) => {

if (err) throw err;

console.log("Completed!");

});

## **fs.writeFile()**

El método fs.writeFile() se utiliza para escribir el contenido en el archivo. Si el archivo no existe en la ruta dada, creará uno nuevo. Pruebe el siguiente código para escribir el contenido en un archivo.

const fs = require("fs");

fs.writeFile("sample.txt", "Writing content", (err) => {

if (err) throw err;

console.log("Completed!");

});

## **fs.readFile()**

El método fs.readFile() se utiliza para leer el contenido de un archivo. Lanzará un error si el archivo no existe en la ruta dada. Examine el siguiente código para el método.

const fs = require("fs");

fs.readFile("sample.txt", (err, data) => {

if (err) throw err;

console.log(data.toString());

});

## **fs.unlink()**

El método fs.unlink() se utiliza para eliminar el archivo. Lanzará un error si el archivo no existe en la ruta dada. Eche un vistazo al código.

const fs = require("fs");

fs.unlink("sample.txt", (err) => {

if (err) throw err;

console.log("File deleted!");

});

## **fs.rename()**

El método fs.rename() se utiliza para cambiar el nombre del archivo. Lanzará un error si el archivo no existe en la ruta dada. Cambie el nombre del siguiente archivo con el siguiente código. ¡Se inteligente!

const fs = require("fs");

fs.rename("sample.txt", "sample\_one.txt", (err) => {

if (err) throw err;

console.log("File renamed!");

});

## **Misceláneo**

Ahora, está familiarizado con diferentes métodos de manejo de archivos de la **fs (sistema de archivos)**módulo. Puede realizar la mayoría de las operaciones de archivo utilizando los métodos que ha visto en este tutorial. Como prometimos, veamos un script de ejemplo que abre un archivo y lee su contenido usando el fs.open() y fs.readFile() métodos respectivamente.

const fs = require("fs");

fs.open("sample.txt", "r", (err, file) => {

if (err) throw err;

fs.readFile(file, (err, data) => {

if (err) throw err;

console.log(data.toString());

});

});

**Conclusión**

Eso es todo por este tutorial. Puede utilizar los métodos de manejo de archivos para automatizar algunas de las cosas aburridas en sus tareas diarias. Espero que haya aprendido los métodos esenciales para manejar archivos.

Codificación feliz 🙂