




Métodos mas comunes de JSON en navegadores JAVASCRIPT

JSON funciona como un **formato de datos universal** que puedes usar para **almacenar, transferir, estructurar y manipular información**, tanto en el navegador como en el backend. Aunque no es un "medio de almacenamiento" por sí solo, **se usa como contenido dentro de medios como** localStorage, **archivos .json**, **APIs**, etc.

Aquí tienes una tabla completa con **funciones útiles de JSON** que puedes aprovechar en tus proyectos:

|  Función |  Descripción |  Ejemplo práctico |
|---|---|--|
| JSON.stringify(obj) | Convierte un objeto JS a cadena JSON | JSON.stringify({nombre: "Alexis"}) |
| JSON.parse(jsonStr) | Convierte una cadena JSON a objeto JS | JSON.parse('{"nombre":"Alexis"}') |
| localStorage.setItem() | Guarda JSON como string en el navegador | localStorage.setItem("usuario", JSON.stringify(obj)) |
| localStorage.getItem() | Recupera JSON y lo convierte a objeto | JSON.parse(localStorage.getItem("usuario")) |
| fetch() con JSON | Envía o recibe JSON desde APIs | fetch("/api", { body: JSON.stringify(data) }) |
| Blob + URL.createObjectURL() | Permite descargar archivos .json | Crear botón para exportar datos como JSON |
| Validación con Zod / Yup | Verifica estructura y tipos de un JSON | Validar datos antes de enviarlos al backend |
| Manipulación con .map(), .filter() | Procesa arrays JSON como datos | usuarios.filter(u => u.activo) |
| Uso en archivos .json | Carga configuraciones, traducciones, etc. | import config from './config.json' |
| Comparación de versiones | Detecta cambios entre dos JSON | Ideal para sincronización de estado |
| Uso como base de configuración | Define rutas, permisos, estilos | menu.json, roles.json, etc. |
| Comunicación entre capas | Transfiere datos entre frontend y backend | React ↔ Spring Boot usando JSON |
| Almacenamiento en IndexedDB | Guarda objetos JSON estructurados | Para apps offline o POS locales |
| Serialización para logs o auditoría | Guarda eventos como JSON | log.push(JSON.stringify(evento)) |