

JSON

JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero y fácil de leer. En JavaScript, JSON se utiliza para representar datos estructurados y para intercambiar datos entre un servidor y un cliente.

1. Sintaxis Básica:

Objeto JSON:

- Se representa entre llaves `{}`.
- Contiene pares clave-valor separados por comas.
- La clave y el valor se separan con dos puntos `:`.

```
{  
  "clave1": "valor1",  
  "clave2": "valor2",  
  "clave3": "valor3"  
}
```

EJEMPLO:

```
{  
  "nombre": "Juan",  
  "edad": 30,  
  "ciudad": "Ejemplo City"  
};
```

2. Tipos de Datos en JSON:

Objeto: Colección no ordenada de pares clave-valor.

```
{  
  "nombre": "Juan",  
  "edad": 30,  
  "ciudad": "Ejemplo City"  
};
```

Array: Lista ordenada de valores.

```
["rojo", "verde", "azul"];
```

String: Cadena de texto.

```
"Hola, mundo!";
```

Número: Números enteros o de punto flotante.

```
42;
```

Booleano: Valores verdadero o falso.

true;

Anidación: Puede anidar objetos y arrays para representar estructuras más complejas.

```
{
  "usuario": {
    "nombre": "María",
    "edad": 25,
    "direcciones": [
      {"calle": "Ejemplo St", "ciudad": "Ejemplo City"},
      {"calle": "Otra St", "ciudad": "Otra City"}
    ]
  }
}
```

3. Acceder a Datos en JSON:

// Acceder a valores en un objeto JSON

console.log(persona.nombre); **Imprime: Juan**

// Acceder a valores en un array JSON

console.log(colores[0]); // **Imprime: rojo**

4. Convertir entre JSON y JavaScript:

JavaScript a JSON:

```
let persona = {
  "nombre": "Juan",
  "edad": 30,
  "ciudad": "Ejemplo City"
};
```

```
let personaJSON = JSON.stringify(persona);
console.log(personaJSON);
```

JSON a JavaScript:

```
let personaJSON = '{"nombre":"Juan","edad":30,"ciudad":"Ejemplo City"}';
let persona = JSON.parse(personaJSON);
console.log(persona.nombre); // Imprime: Juan
```

5. Acceso y escritura

Acceso a JSON y tratamiento de datos. EJERCICIO PRUEBA 1:

// Importa el módulo 'fs' (File System) de Node.js

```
const fs = require('fs');
```

// Lee el archivo JSON

```
fs.readFile('datos.json', 'utf8', (err, data) => {  
  if (err) {  
    console.error('Error al leer el archivo:', err);  
    return;  
  }  
}
```

// Convierte el contenido del archivo JSON a un objeto JavaScript

```
const datos = JSON.parse(data);
```

// Accede a los valores del objeto JSON

```
console.log('Usuarios:');  
datos.usuarios.forEach(usuario => {  
  // Muestra información de cada usuario en la consola  
  console.log(`ID: ${usuario.id}, Nombre: ${usuario.nombre}, Edad: ${usuario.edad}`);  
});
```

Escribir datos en JSON. EJERCICIO PRUEBA 2:

```
const fs = require('fs');
```

// Datos a escribir en el archivo JSON

```
const nuevoDatos = {  
  usuarios: [  
    { id: 1, nombre: "NuevoUsuario1", edad: 25 },  
    { id: 2, nombre: "NuevoUsuario2", edad: 30 },  
    { id: 3, nombre: "NuevoUsuario3", edad: 22 }  
  ]  
};
```

// Convierte el objeto JavaScript a una cadena JSON

```
const nuevoDatosJSON = JSON.stringify(nuevoDatos, null, 2);
```

// Escribe la cadena JSON en el archivo

```
fs.writeFile('nuevo_datos.json', nuevoDatosJSON, 'utf8', (err) => {  
  if (err) {  
    console.error('Error al escribir en el archivo:', err);  
    return;  
  }  
  console.log('Datos escritos exitosamente en nuevo_datos.json');  
});
```

EJERCICIO JSON.

Crea una página en la que se lea un archivo JSON existente, se añada un nuevo elemento y finalmente, se escriba en un nuevo archivo JSON.

1. Lee el archivo JSON existente (por ejemplo, datos.json, en el que haya varios usuarios con al menos 3 datos).
2. Convierte el contenido a un objeto JavaScript.
3. Modifica los datos agregando un nuevo usuario.
4. Convierte el objeto JavaScript modificado a una cadena JSON.
5. Escribe la cadena JSON en un nuevo archivo (datos_modificados.json).
- * 6. Repite el proceso anterior, pero ahora escribe los nuevos usuarios sobre el mismo fichero inicial (datos.json).