

JSON

JA_V_A_S_C_R_I_P_T

OB_J_E_C_T

NO_T_A_T_I_O_N



**Certified Tech
Developer**
The Ultimate Degree



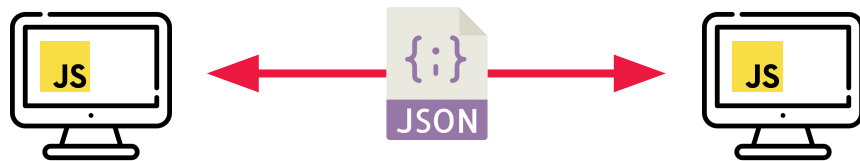
Es un **formato** de **texto** sencillo
utilizado para el **intercambio** de **datos**
entre distintos **sistemas**.



¿Por qué es tan popular JSON?

En la web, la mayoría de las peticiones y sus respuestas viajan como texto plano, es decir, texto sin codificaciones especiales.

JSON, al ser una **cadena de texto** simple, es un **formato ideal para transmitir información** entre sitios y aplicaciones web. Especialmente si tenemos en cuenta que JavaScript está presente en todos los navegadores modernos.



La otra ventaja de JSON es que cualquier lenguaje de programación puede interpretarlo con facilidad. De hecho, la mayoría de los lenguajes web trabajan nativamente con JSON.

El objeto literal y JSON – Estructura

JSON es el acrónimo de JavaScript Object Notation y, como su nombre lo indica, es muy similar al objeto literal que ya conocemos. Veamos las diferencias:

| Objeto Literal | JSON |
|--|--|
| Admite comillas simples y dobles. | Solo se pueden usar comillas dobles. |
| Las claves del objeto van sin comillas. | Las claves van entre comillas. |
| Podemos escribir métodos sin problemas. | No admite métodos, solo propiedades y valores. |
| Se recomienda poner una coma en la última propiedad. | No podemos poner una coma en el último elemento. |

El objeto literal y JSON – Código

JSON admite la mayoría de los tipos de datos de JavaScript. Veamos cómo sería la conversión entre ambos formatos.

JS

```
{  
  texto: 'Mi texto',  
  numero: 16,  
  array: ['uno', 'dos'],  
  booleano: true,  
  metodo(): {return '¡Hola!'},  
}
```



{JSON}
JavaScript Object Notation

```
{  
  "texto": "Mi texto",  
  "numero": 16,  
  "array": ["uno", "dos"],  
  "booleano": true  
}
```

JSON no soporta métodos. ⚠

“

JavaScript nos proporciona un **objeto nativo JSON** con dos métodos que **nos permiten convertir el formato** de un archivo JSON a objeto literal o array —y viceversa—.

”



JSON.parse()

Convierte un texto con formato JSON al tipo de dato equivalente de JavaScript.

- **Recibe** una cadena de texto con formato **JSON**.
- **Devuelve** el mismo dato que recibió en formato **JavaScript**.

```
{}
```

```
let datosJson = '{"club": "Independiente", "barrio": "Avellaneda"}';  
let datosConvertidos = JSON.parse(datosJson);  
  
console.log(datosConvertidos);  
// Se verá en consola un objeto literal  
// { club: 'Independiente', barrio: 'Avellaneda' }
```

JSON.stringify()

Convierte un tipo de dato de JavaScript en un texto en formato JSON.

- **Recibe** un tipo de dato de **JavaScript**.
- **Devuelve** una cadena de texto con formato **JSON**.

```
{  
  let objeto Literal = { nombre: 'América de Cali', pais: 'Colombia' };  
  let datosConvertidos = JSON.stringify(objetoLiteral);  
  
  console.log(datosConvertidos);  
  // Se verán en consola los datos en un string de tipo JSON  
  // '{ "nombre": "América de Cali", "pais": "Colombia" }'
```


DigitalHouse>
Coding School