

JSON

JAVASCRIPT OBJECT NOTATION

“

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

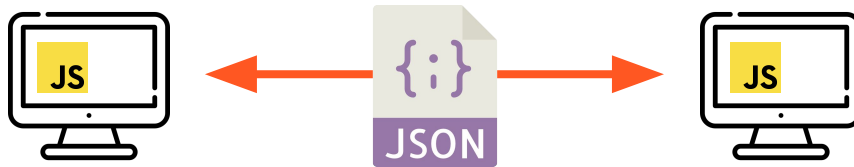


LA POPULARIDAD DE JSON

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.

ESTRUCTURA JSON

Como su nombre lo indica, **JSON** es muy similar a un **objeto literal**. La diferencias entre ellos son:

Objeto Literal

- Admite comillas simples y dobles
- Las claves del objeto van sin comillas
- Podemos escribir métodos
- Se recomienda poner una coma en la última propiedad

JSON

- Sólo se pueden usar comillas dobles
- Las claves van entre comillas
- No admite métodos, sólo propiedades y valores
- No podemos poner una coma en el último elemento

ESTRUCTURA JSON

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 JSON al tipo de dato equivalente de Javascript

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

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

```
let objetoLiteral = { nombre: 'Carla', pais: 'Argentina' };  
let datosConvertidos = JSON.stringify(datosObjeto);
```

```
console.log(datosConvertidos);  
// Se verán en consola los datos en un string de tipo JSON  
// '{ "nombre": "Carla", "pais": "Argentina" }'
```

{}

Gracias a estos dos métodos
podremos generar un
formato transaccional de
fácil comprensión **entre**
distintos **sistemas**.

