

ARCHIVO EN FORMATO JSON (JavaScript Object Notation)

JSON es un formato que almacena información estructurada y se utiliza principalmente para transferir datos entre un servidor y un cliente.

El archivo es básicamente una alternativa más simple y liviana al XML (Lenguaje de marcado extenso, por sus siglas en inglés) que cuenta con funciones similares.

Los desarrolladores usan JSON para trabajar con [AJAX](#) (JavaScript asíncrono y XML, por sus siglas en inglés). Estos formatos funcionan bien juntos para lograr la carga asincrónica de los datos almacenados, lo que significa que un sitio web puede actualizar su información sin actualizar la página.

Este proceso es más fácil de hacer con JSON que con XML/RSS. Y hoy, como muchos sitios web están adoptando AJAX, el archivo **.json** se ha vuelto muy popular.

Además, permite a los usuarios solicitar datos de un [dominio](#) diferente con un método llamado [JSONP](#) mediante la aplicación de etiquetas **<script>**. De lo contrario, no puedes transferir dominios cruzados de datos debido a la política "[mismo-origen](#)" (same-origin).

Sintaxis JSON

Ahora que sabemos lo que es un archivo JSON, es importante saber la sintaxis correcta.

Hay dos elementos centrales en un objeto JSON: claves (Keys) y valores (Values).

- Las **Keys** deben ser cadenas de caracteres (strings). Como su nombre en español lo indica, estas contienen una secuencia de caracteres rodeados de comillas.
- Los **Values** son un tipo de datos JSON válido. Puede tener la forma de un arreglo (array), objeto, cadena (string), booleano, número o nulo.

Un objeto JSON comienza y termina con llaves { }. Puede tener dos o más pares de **claves/valor** dentro, con una **coma** para separarlos. Así mismo, cada key es seguida por **dos puntos** para distinguirla del valor.

Como resultado, la sintaxis literal del objeto JSON se ve así:

```
{“key”:“value”,“key”:“value”,“key”:“value”.}
```

Tipos de valores

Como se mencionaba antes, los valores contienen un tipo de datos JSON válido, como:

Array

Un array (en español conocido como arreglo o vector) es una colección ordenada de valores. Está rodeado de **corchetes** `[]` y cada valor dentro está separado por una coma.

Un valor de un array puede contener objetos JSON, lo que significa que utiliza el mismo concepto de par clave/valor. Por ejemplo:

```
"estudiantes": [  
  {"primerNombre":"Tom", "Apellido":"Jackson"},  
  {"primerNombre":"Linda", "Apellido":"Garner"},  
  {"primerNombre":"Adam", "Apellido":"Cooper"}  
]
```

En este caso, la información entre corchetes es un array que tiene tres objetos.

Objeto

Un objeto contiene una clave y un valor. Hay dos puntos después de cada clave y una coma después de cada valor, que también distingue a cada objeto. Ambos están entre comillas.

El objeto, como valor, debe seguir la misma regla que un objeto común. Ejemplo:

```
{  
  "empleados":{  
    "nombre":"Tom",  
    "apellido":"Jackson"  
  }  
}
```

Aquí, **empleados** es la clave, mientras que todo lo que está dentro de las llaves es el objeto.

Strings

Un string (conocido en español como cadena de caracteres) es una secuencia establecida de cero o más caracteres Unicode. Está encerrado entre dos comillas dobles.

Este ejemplo muestra que **Tom** es un string ya que es un conjunto de caracteres dentro de una comilla doble.

```
"Primer Nombre":"Tom"
```

Número

El número en JSON debe ser un **número entero** o un **punto flotante**, como

```
{"Edad": "30"}
```

Booleano

Puedes usar **verdadero** o **falso** como valor, de la siguiente manera:

```
{"Casado": "false"}
```

Nulo

Es para mostrar que no hay información.

```
{"Tipo de sangre": "null"}
```

Datos JSON almacenados

Continuando con esta explicación acerca de qué es JSON, pasemos a hablar de que hay dos formas de almacenar datos JSON: objeto y vector. El primero se ve así:

```
{  
  "nombre": "Tom",  
  "apellido": "Jackson",  
  "género": "masculino"  
}
```

Las llaves indican que es un objeto JSON. Implica tres pares clave/valor que están separados por comas.

En cada par, tienes las claves (nombre, apellido y género) seguidas de dos puntos para distinguirlos de los valores (Tom, Jackson, masculino).

Los valores en este ejemplo son strings. Por eso también están entre comillas, similares a las claves.

Usando vectores

Otro método para almacenar datos es un vector (array). Échale un vistazo a éste ejemplo:

```
{  
  "nombre": "Tom",  
  "apellido": "Jackson",  
  "género": "masculino",  
  "hobby": ["fútbol", "lectura", "natación"]  
}
```

Lo que diferencia esto del método anterior es el cuarto par clave/valor. **Hobby** es la clave y hay varios valores (fútbol, lectura, natación) entre corchetes, que representan un vector.

Puede ser útil cuando se combina con JSONP para superar el problema entre dominios. Este proceso funciona utilizando lo que se denomina devoluciones de llamada (**callbacks**), que solicitarán un elemento específico del vector sin obtener un error “del mismo origen” (same-origin).

Y afortunadamente, un Array también admite [bucles](#), lo que te permite ejecutar comandos repetidos para buscar múltiples datos, haciendo que el proceso sea más rápido y efectivo.

MÉTODOS DE CONVERSIÓN

JSON.stringify()

El método JSON.stringify() convierte un objeto o valor de JavaScript en una cadena JSON, reemplazando opcionalmente valores si se especifica una función de reemplazo u opcionalmente incluyendo solo las propiedades especificadas si se especifica una matriz de reemplazo.

JSON.parse()

El análisis JSON es el proceso de convertir un objeto JSON (el cual está en formato de texto) en un objeto Javascript que se puede usar dentro de un programa. En Javascript, la forma estándar de hacer esto es usando el método JSON.parse(), como especifica el estándar de Javascript.

EJEMPLO DE JSON CON JAVASCRIPT

```
<html>
<head>
<title> JSON con JavaScript </title>
<script>
  var Libro1 = {"Lenguaje" : "Java", "Autor" : "Hebert Schildt" };
  document.write("<h1>Ejemplo JSON con JavaScript </h1>");

  document.write("<h3> Lenguaje : " + Libro1.Lenguaje + "</h3>");
  document.write("<h3> Autor : " + Libro1.Autor + "</h3>");

  var Libro2 = {"Lenguaje" : "C++", "Autor" : "Deitel"};

  document.write("<h3> Lenguaje : " + Libro2.Lenguaje + "</h3>");
  document.write("<h3> Autor : " + Libro2.Autor + "</h3>");

  document.write(Libro2.Lenguaje + " es un lenguaje de programación que puede ser
estudiado " + "del libro escrito por " + Libro2.Autor);
  document.write("<br>");

</script>
</head>
<body>
</body>
</html>
```

Al ejecutarse el archivo en el navegador aparecerá:

Ejemplo JSON con JavaScript

Lenguaje : Java

Autor : Hebert Schildt

Lenguaje : C++

Autor : Deitel

C++ es un lenguaje de programación que puede ser estudiado del libro escrito por Deitel

Conclusión

Como puedes ver, se trata de una herramienta útil para intercambiar datos. Tiene muchas ventajas:

- Puedes cargar información de forma asíncrona para que tu sitio web responda mejor y pueda manejar el flujo de datos con mayor facilidad.
- También puedes usarlo para superar problemas de dominio cruzado al intercambiar datos desde otro sitio.
- Un archivo JSON es más simple y más liviano que un archivo XML.

REFERENCIAS:

<https://www.hostinger.mx/tutoriales/que-es-json>

<https://www.w3schools.com/>

<https://www.udacity.com/blog/2021/02/javascript-json-parse.html>

<https://platzi.com/clases/2419-javascript-poo-intermedio/39813-jsonparse-y-jsonstringify/#:~:text=stringify%20permite%20volver%20un%20objeto,podemos%20clonar%20un%20objeto%20JavaScript.>