

DWEC – Javascript Web Cliente.

JavaScript – Formato JSON	1
Introducción.....	1
Características del formato	1
Ejemplos.....	1
Estructura de los datos	3
Métodos del objeto JSON	3

JavaScript – Formato JSON

Introducción

El formato *JSON* (acrónimo de **JavaScript Object Notation**, 'notación de objeto de JavaScript') es una forma de **convertir objetos Javascript en una cadena de texto para poderlos enviar.**

Características del formato

- La **estructura** de datos JSON se compone de un **conjunto de objetos o arrays** que **contendrán números, cadenas booleanos y nulos.**
- Un **objeto** JSON **comienza y termina con llaves**, y **contiene** una **colección desordenada de pares nombre-valor.**
- Cada **nombre y valor** están **separados por dos puntos**, y **los pares están separados por comas.**
- La **coma final** está prohibida.
- El **nombre** es una **cadena entre comillas dobles**. Los caracteres de comillas no deben ser inclinadas o "inteligentes".
- En los **números**, los **ceros a la izquierda están prohibidos**; un punto decimal debe estar seguido al menos por un dígito.
- En las **cadenas** deben estar **entre comillas dobles**. No se permiten todos los caracteres de escape; sí se permiten los caracteres de separador de línea Unicode (U+2028) y el separador de párrafo (U+2029).
- Un array JSON comienza y termina con corchetes y contiene una colección ordenada de valores separados por comas. Un valor puede ser una cadena entre comillas dobles, un número, un booleano true o false, nulo, un objeto JSON o un array.
- Los objetos y los arrays JSON se pueden anidar, lo que posibilita una estructura jerárquica de datos.

Ejemplos

En el siguiente ejemplo, se muestra una estructura de datos JSON con dos objetos válidos.

```
{  
  "id": 1006410,  
  "title": "Amazon Redshift Database Developer Guide"  
}
```

```
{  
  "id": 100540,  
  "name": "Amazon Simple Storage Service User Guide"  
}
```

En el siguiente se muestran los mismos datos como dos arrays JSON:

```
[  
  1006410,  
  "Amazon Redshift Database Developer Guide"  
]  
[  
  100540,  
  "Amazon Simple Storage Service User Guide"  
]
```

Conversiones de objetos

El objeto alumno:

```
let alumno = {  
  id: 5,  
  nombre: 'Ana',  
  apellidos: 'Zubiri Peláez'  
}
```

se transformaría en la cadena de texto:

```
{ "id": 5, "nombre": "Ana", "apellidos": "Zubiri Peláez" }
```

y el array:

```
let alumnos = [  
  {  
    id: 5,  
    nombre: "Ana",  
    apellidos: "Zubiri Peláez"  
  },  
  {  
    id: 7,  
    nombre: "Carlos",  
    apellidos: "Pérez Ortíz"  
  },  
]
```

Se transformaría en la cadena:

```
[{ "id": 5, "nombre": "Ana", "apellidos": "Zubiri Peláez" }, { "id": 7, "nombre":  
  "Carlos", "apellidos": "Pérez Ortíz" }]
```

Nótese que tanto las claves como los valores van entrecomillados (con comillas dobles). No sirven comillas simples.

Estructura de los datos

Los mismos datos pueden tener distinta estructura.

Archivo colores1.json	Archivo colores2.json	Archivo colores3.json
<pre>{ "arrayColores":[{ "nombreColor":"rojo", "valorHexadec":"#f00" }, { "nombreColor":"verde", "valorHexadec":"#0f0" }, { "nombreColor":"azul", "valorHexadec":"#00f" }, { "nombreColor":"cyan", "valorHexadec":"#0ff" }, { "nombreColor":"magenta", "valorHexadec":"#f0f" }, { "nombreColor":"amarillo", "valorHexadec":"#ff0" }, { "nombreColor":"negro", "valorHexadec":"#000" }] }</pre>	<pre>{ "arrayColores":[{ "rojo":"#f00", "verde":"#0f0", "azul":"#00f", "cyan":"#0ff", "magenta":"#f0f", "amarillo":"#ff0", "negro":"#000" }] }</pre>	<pre>{ "rojo":"#f00", "verde":"#0f0", "azul":"#00f", "cyan":"#0ff", "magenta":"#f0f", "amarillo":"#ff0", "negro":"#000" }</pre>

Los ejemplos anteriores representan lo que podrían ser archivos JSON conteniendo datos en formato JSON.

Se trata de 3 archivos que contienen aproximadamente la misma información. Sin embargo, hay algunas diferencias:

- En el archivo **colores1.json** existe un único objeto de datos donde el nombre es *arrayColores* y su valor es un array de objetos JSON. Cada objeto del array está formado por los pares (*nombreColor* y su valor), y (*valorHexadec* y su valor). En este ejemplo en concreto el array consta de 7 elementos con información correspondiente a 7 colores.
- En el archivo **colores2.json** existe un único objeto de datos donde el nombre es *arrayColores*, cuyo valor es un array que contiene un único objeto JSON formado por siete pares (nombre – valor) que representa información sobre siete colores.
- En el archivo **colores3.json** existe un único objeto de datos que está formado por siete pares (nombre – valor) que representa información sobre siete colores.

Siendo las 3 formas válidas, se deberá utilizar aquella que se indique en las instrucciones o, de no existir pautas precisas, utilizar aquel diseño que favorezca el desarrollo y mantenimiento de la aplicación.

Métodos del objeto JSON

Para convertir objetos en cadenas de texto *JSON* y viceversa, Javascript proporciona 2 métodos:

- **JSON.stringify(objeto)**: recibe un objeto JS y devuelve la cadena de texto correspondiente.

```
//convierte de formato JSON a objeto
const cadenaAlumnos = JSON.stringify(alumnos)
```

- **JSON.parse(cadena)**: realiza el proceso inverso, convirtiendo una cadena de texto en un objeto.

```
//convierte un objeto a formato JSON
```

```
const alumnos = JSON.parse(cadenaAlumnos)
```