

# UD1. JSON

## Índice

Introducción.....	2
Ejemplo de JSON.....	2
¿Por qué utilizar JSON?.....	2
Sintaxis JSON.....	3

# Introducción

JSON significa Notación de Objetos de Java Script. La sintaxis JSON se deriva de la notación de objetos de JavaScript, pero el formato JSON es solo texto.

JSON es un formato de texto para almacenar y transportar datos.

JSON es "autodescriptivo" y fácil de entender

JSON es un formato liviano de intercambio de datos

JSON es texto simple escrito en notación de objetos JavaScript

JSON se utiliza para enviar datos entre computadoras

El tipo de archivo para los archivos JSON es ".json"

El tipo MIME para el texto JSON es "application/json"

## Ejemplo de JSON

**Este ejemplo es una cadena JSON: '{ "name" : "John" , "age" : 30 , "car" : null }'**

Define un objeto con 3 propiedades para definir : nombre, edad , coche

Cada propiedad tiene un valor.

Si analiza la cadena JSON con un programa JavaScript, puede acceder a los datos como un objeto y por tanto usar el operador punto (.) para acceder a sus propiedades:

```
let personName = obj.name;
```

## ¿Por qué utilizar JSON?

El formato JSON es sintácticamente similar al código para crear objetos JavaScript. Por ello, un programa JavaScript puede convertir fácilmente datos JSON en objetos JavaScript.

Dado que el formato es solo texto, los datos JSON pueden enviarse fácilmente entre computadoras y ser utilizados por cualquier lenguaje de programación.

JavaScript tiene una función incorporada para convertir cadenas JSON en objetos JavaScript:

JSON.parse()

JavaScript también tiene una función incorporada para convertir un objeto en una cadena JSON:

JSON.stringify()

Puede recibir texto puro de un servidor y usarlo como un objeto JavaScript.

Puede enviar un objeto JavaScript a un servidor en formato de texto puro.

Puede trabajar con datos como objetos JavaScript, sin análisis ni traducciones complicados.

Almacenamiento de datos

Al almacenar datos, estos deben tener un formato determinado y, sin importar dónde elija almacenarlos, el texto siempre es uno de los formatos legales.

JSON permite almacenar objetos JavaScript como texto.

# Sintaxis JSON

La sintaxis JSON se deriva de la sintaxis de notación de objetos de JavaScript:

Los datos están en pares clave/valor

Los datos están separados por comas.

Las llaves sostienen objetos

Los corchetes contienen matrices

Datos JSON: un nombre y un valor

Un par nombre/valor consta de un nombre de campo (entre comillas dobles), seguido de dos puntos, seguido de un valor:

Los nombres JSON requieren comillas dobles.

En JSON , los valores deben ser uno de los siguientes tipos de datos:

una cadena de caracteres (irá entre comillas dobles también) : **{"name" : "John"}**

Un numero : **{"age" : 23 }**

Un objeto : **{ "employee":{"name":"John", "age":30,"city":"New York"} }**

una matriz : **{ "employees":["John","Anna","Peter"] }**

un booleano : **{ "isStudent" : true }**

nulo : **{ "car" : null }**

Con JavaScript puedes crear un objeto y asignarle datos, de la siguiente manera:

Ejemplo

```
person = {name:"John", age:31, city:"New York"};
```

Puedes acceder a un objeto JavaScript de la siguiente manera:

Ejemplo

```
// devuelve John
```

```
person.name;
```

También se puede acceder de esta manera:

```
person["name"];
```

Los datos se pueden modificar de la siguiente manera:

```
person.name = "Gilbert";
```

También se puede modificar así:

```
person["name"] = "Gilbert";
```

JSON VS XMLSe pueden utilizar tanto JSON como XML para recibir datos de un servidor web.

Los siguientes ejemplos JSON y XML definen un objeto de empleados, con una matriz de 3 empleados:

Ejemplo de JSON	Ejemplo de XML
<pre>{   "employees": [     { "firstName": "John", "lastName": "Doe" },     { "firstName": "Anna", "lastName": "Smith" },     { "firstName": "Peter", "lastName": "Jones" }   ] }</pre>	<pre>&lt;employees&gt;   &lt;employee&gt;     &lt;firstName&gt;John&lt;/firstName&gt;     &lt;lastName&gt;Doe&lt;/lastName&gt;   &lt;/employee&gt;   &lt;employee&gt;     &lt;firstName&gt;Anna&lt;/firstName&gt;     &lt;lastName&gt;Smith&lt;/lastName&gt;   &lt;/employee&gt;   &lt;employee&gt;     &lt;firstName&gt;Peter&lt;/firstName&gt;     &lt;lastName&gt;Jones&lt;/lastName&gt;   &lt;/employee&gt; &lt;/employees&gt;</pre>

Tanto JSON como XML son "autodescriptivos" (legibles para humanos)

Tanto JSON como XML son jerárquicos (valores dentro de valores)

Tanto JSON como XML pueden analizarse y utilizarse en muchos lenguajes de programación.

Tanto JSON como XML se pueden obtener con XMLHttpRequest

JSON es diferente a XML porque

- JSON no utiliza etiqueta final

- JSON es más corto

- JSON es más rápido de leer y escribir

- JSON puede utilizar matrices

La mayor diferencia es: que El XML debe analizarse con un analizador XML. JSON no.

Para aplicaciones AJAX, JSON es más rápido y sencillo que XML