

JSON(JavaScript Object Notation)

{JSO^oN}

JSON(JavaScript Object Notation)

- ▶ JSON es un formato para guardar e intercambiar información que cualquier persona pueda leer. Los archivos contienen solo texto y usan la extensión “.json”.
- ▶ Este formato almacena información estructurada y se utiliza principalmente para transferir datos entre un servidor y un cliente, el formato JSON es básicamente una alternativa más simple y liviana al XML(Lenguaje de marcado extenso).

¿Cómo surgió JSON?

- ▶ A principios de la década de los 90 surgió el problema de que las máquinas pudieran entenderse entre sí e intercambiar información, una solución fue crear el estándar XML, sin embargo, XML presentaba problemas sobre todo cuando se trataba de trabajar con gran volumen de datos, puesto que el procesamiento de los datos en este formato se volvía lento.
- ▶ Surgieron entonces intentos para definir formatos que fueran más ligeros y rápidos para el intercambio de información uno de ellos fue JSON, promovido y popularizado a principios de los 2000 por Douglas Crockford, un programador conocido como el 'gurú' de JavaScript.

Características

- ▶ Json es solo un formato de datos.
- ▶ Requiere usar comillas dobles para las cadenas y los nombres de las propiedades. Las comillas simples no son válidas.
- ▶ Una coma o dos puntos mal ubicados pueden producir que un archivo JSON no funcione.
- ▶ Puede tomar la forma de cualquier tipo de datos que sea válido para ser incluido en un JSON, no solo arreglos u objetos. Así, por ejemplo, una cadena o un número único podrían ser objetos JSON válidos.
- ▶ A diferencia del código JavaScript, en el que las propiedades del objeto pueden no estar entre comillas, en JSON solo las cadenas entre comillas pueden ser utilizadas como propiedades.

Ventajas de JSON

- ▶ Es auto descriptivo y fácil de entender.
- ▶ Su sencillez le ha permitido posicionarse como alternativa a XML.
- ▶ Es más rápido en cualquier navegador.
- ▶ Es más fácil de leer que XML.
- ▶ Es más ligero(bytes) en las transmisiones.
- ▶ Se parsea más rápido.
- ▶ Velocidad de procesamiento rápida.
- ▶ Puede ser entendido de forma nativa por los analizadores de JavaScript.

Sintaxis

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

- **Keys:**

Las keys deben ser cadenas de caracteres(string). Como su nombre lo indica, estas contienen una secuencia de caracteres rodeadas entre comillas “” las cuales identifican un determinado valor.

- **Values:**

Los values representan un valor, pueden tener la forma de un arreglo(array), objeto, cadena(string), booleando, número o nulo.

Ejemplo:

```
{  
  "lenguaje": "Java",  
  "paradigma": "POO"  
}
```

Tipos de valores

► 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 de un objeto común. Ejemplo:

```
{  
  "lenguaje": {  
    "nombre": "Java",  
    "paradigma": "POO"  
  }  
}
```

En este caso, “lenguaje” es la clave, mientras que todo lo que está dentro de las llaves es el objeto.

Tipos de valores

► Strings

Un string es una secuencia establecida de cero o más caracteres Unicode. Está cerrada entre dos comillas dobles.

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

```
"nombre": "Java"
```


Tipos de valores

► Número

El número en JSON debe ser un número entero o punto flotante, como se muestra a continuación(Las comillas dobles son opcionales).

```
"edad": 24
```

Tipos de valores

► Booleano

Puedes usar **verdadero(true)** o **falso(false)** como valor, de la siguiente manera (Las comillas dobles son opcionales).

```
"pagado": false,  
"activo": true
```

Tipos de valores

► Nulo

Para indicar que no hay información, es decir valor null, se hace cómo se muestra a continuación (Las comillas dobles son opcionales).

```
"segundoApellido": null,
```

Tipos de valores

► Array

Un array (en español conocido como arreglo o vector) es una colección ordenada de valores.

Está rodeada por corchetes “[]” y cada valor dentro está separado por una coma.

Un valor de array puede obtener cualquiera de los valores antes vistos, lo que significa que utiliza el mismo concepto de par clave/valor.

```
{
  "lenguajes": [
    {
      "nombre": "Java",
      "paradigma": "POO"
    },
    {
      "nombre": "Python",
      "paradigma": "Multiparadigma"
    },
    {
      "nombre": "C#",
      "paradigma": "POO"
    },
    {
      "nombre": "Clojure",
      "paradigma": "Funcional"
    }
  ]
}
```

Ejemplos

```
{
  "marcadores": [
    {
      "latitude": 40.416875,
      "longitude": -3.703308,
      "city": "Madrid",
      "description": "Puerta del Sol"
    },
    {
      "latitude": 40.417438,
      "longitude": -3.693363,
      "city": "Madrid",
      "description": "Paseo del Prado"
    },
    {
      "latitude": 40.407015,
      "longitude": -3.691163,
      "city": "Madrid",
      "description": "Estación de Atocha"
    }
  ]
}
```