



iesperemariaorts

JSON

JavaScript Object Notation - Notación de Objetos de JavaScript

Formato ligero de almacenamiento/intercambio de datos

Leerlo y escribirlo es simple para humanos y máquinas (**autodescriptivo**).

Es completamente **independiente del lenguaje** - utiliza convenciones que son ampliamente conocidas por los programadores, por ello es un lenguaje ideal para el intercambio de datos.



iesperemariaorts

JSON

Sintaxis JSON, 5 reglas:

- Es un subconjunto de la sintaxis de JavaScript.
- Los datos aparecen como pares nombre/valor.
 - Los datos se separan con comas.
 - Las llaves { } contienen objetos.
 - Los corchetes [] contienen arrays.



iesperemariaorts

JSON

Valores en un JSON

- Un número (integer or floating point)
- Una cadena de texto (in double quotes)
- Un Booleano (true o false)
- Un array (in [])
- Un objeto (in {})
- NULL



iesperemariaorts

JSON

Parecido al XML

- Tanto el JSON como el XML son auto-descriptivos
- Tanto el JSON como el XML son jerárquicos (valores dentro de valores)
- Tanto el JSON como el XML pueden ser «parseados» por muchos lenguajes de programación
- Tanto el JSON como el XML pueden ser recuperados a través del XHR



iesperemariaorts

JSON

Y a diferencia del XML

- JSON no utiliza tags de finalización
- JSON es más corto
- JSON es más rápido de leer y escribir
- JSON puede utilizar arrays
- La principal diferencia es que el XML tiene que ser parseado con un XML parser mientras que el JSON puede ser parseado por cualquier función en Javascript



iesperemariaorts

JSON

Usos

- Un uso típico de JSON es para leer datos de un servidor y mostrar esos datos en una página web
- Veamos un ejemplo en el que creamos un string en Javascript con notación/sintaxis de JSON
- Usaremos la función `JSON.parse(text)` para convertir texto en JSON a un objeto en Javascript:



iesperemariaorts

JSON

JSON.parse(text) – Ejemplo recepción datos de un servidor:

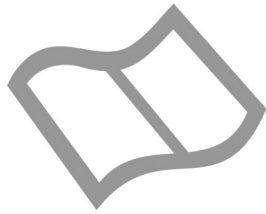
Recibimos del servidor estos datos:

```
{ "name": "John", "age": 30, "city": "New York" }
```

Los "parseamos", convirtiéndolos a objetos de Javascript

```
var obj = JSON.parse('{ "name": "John", "age": 30, "city": "New York" }');
```

Accedemos como objeto: obj.name, obj.age;



iesperemariaorts

JSON

```
var text = '{ "employees" : [' +  
  '{ "firstName":"John" , "lastName":"Doe" },' +  
  '{ "firstName":"Anna" , "lastName":"Smith" },' +  
  '{ "firstName":"Peter" , "lastName":"Jones" } ]}';
```

```
var obj = JSON.parse(text);
```

```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML =
```

```
obj.employees[1].firstName + " " + obj.employees[1].lastName;
```

```
</script>
```




iesperemariaorts

JSON

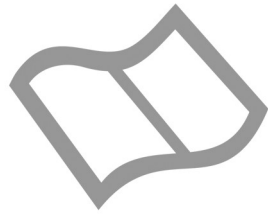
JSON.stringify() – Realiza el proceso inverso a JSON.parse()

```
1 console.log(JSON.stringify({ x: 5, y: 6 }));  
2 // expected output: '{"x":5,"y":6}'  
3  
4 console.log(JSON.stringify([new Number(3), new String('false'), new Boolean(false)]));  
5 // expected output: '[3,"false",false]'  
6  
7 console.log(JSON.stringify({ x: [10, undefined, function() {}, Symbol('')] }));  
8 // expected output: '{"x":[10,null,null,null]}'  
9  
10 console.log(JSON.stringify(new Date(2006, 0, 2, 15, 4, 5)));  
11 // expected output: '"2006-01-02T15:04:05.000Z"'  
12
```

Run ›

Reset

```
> '{"x":5,"y":6}'  
> '[3,"false",false]'  
> '{"x":[10,null,null,null]}'  
> '"2006-01-02T14:04:05.000Z'"
```



iesperemariaorts

JSON

JSON.stringify() – Convierte un objeto en formato JSON
(texto)

JSON.parse() – Convierte el formato JSON (texto) en un
objeto

**SE DEBEN DE UTILIZAR ESTOS MÉTODOS PARA
TRANSFORMAR LOS DATOS. NUNCA DEBEMOS
HACERLO NOSOTROS "MANUALMENTE"**