

# Formatos para envío y recepción de la información

## JSON



**Formato para el intercambio de datos.**

**Alternativa a XML**

**Fácil uso en Javascript.**

**Puede ser leído por cualquier lenguaje de programación**



# Formatos para envío y recepción de la información

## JSON

### Nombre/Par de valores

`"Nombre" : "Chema"`

### Valores

**Número (entero o float)**

**String (entre comillas simples)**

**Booleano (true o false)**

**Array (entre corchetes)**

**Objeto (entre llaves)**

**Null**

### Objetos

```
{ "NombreFruta": "Manzana", "Cantidad": 20 }
```

### Arrays

```
{  
  "Frutas": [  
    { "NombreFruta": "Manzana", "cantidad": 10 },  
    { "NombreFruta": "Pera", "cantidad": 20 },  
    { "NombreFruta": "Naranja", "cantidad": 30 }  
  ]  
}
```



# Formatos para envío y recepción de la información

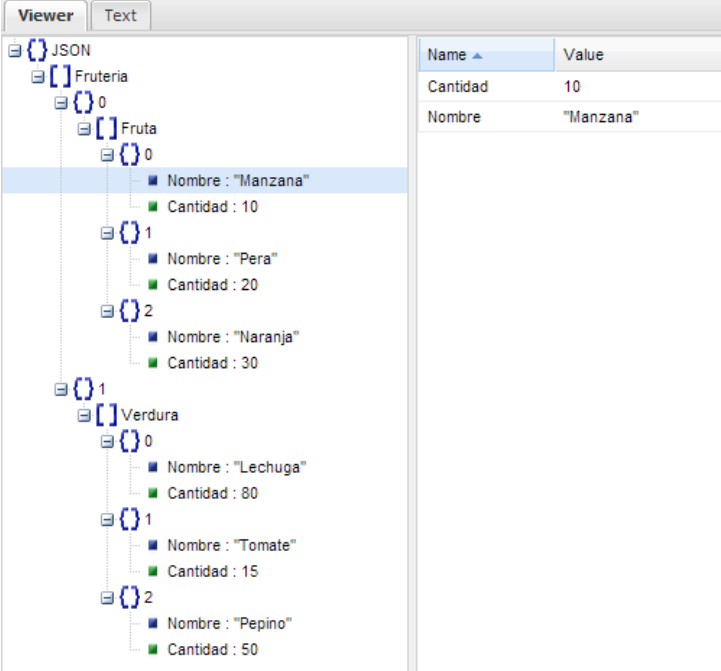
## JSON

```
{
  "Fruteria": [
    {
      "Fruta": [
        { "Nombre": "Manzana", "Cantidad": 10 },
        { "Nombre": "Pera", "Cantidad": 20 },
        { "Nombre": "Naranja", "Cantidad": 30 }
      ]
    },
    {
      "Verdura": [
        { "Nombre": "Lechuga", "Cantidad": 80 },
        { "Nombre": "Tomate", "Cantidad": 15 },
        { "Nombre": "Pepino", "Cantidad": 50 }
      ]
    }
  ]
}
```

# Formatos para envío y recepción de la información

**JSON**

```
{
  "Fruteria": [
    {
      "Fruta": [
        { "Nombre": "Manzana", "Cantidad": 10 },
        { "Nombre": "Pera", "Cantidad": 20 },
        { "Nombre": "Naranja", "Cantidad": 30 }
      ]
    }
  ],
  "Verdura": [
    { "Nombre": "Lechuga", "Cantidad": 80 },
    { "Nombre": "Tomate", "Cantidad": 15 },
    { "Nombre": "Pepino", "Cantidad": 50 }
  ]
}
```



Name	Value
Cantidad	10
Nombre	"Manzana"

# Formatos para envío y recepción de la información

## Recepción de datos en formato JSON

- En la función `iniciar()` solicitamos que cargue, de manera asíncrona, el fichero PHP que devuelve los datos XML. Para ello usamos:  

```
cargarAsync("fichero.php");
```
- Al recibir datos en formato JSON debemos hacer la solicitud con la propiedad `.responseText` del objeto XHR y la parseamos con `JSON.parse` teniendo en cuenta que devolverá un array.  

```
resultados = JSON.parse(this.responseText);
```
- En la función `estadoPetición` realizamos el procesamiento del JSON. Para el ejemplo de las frutas:  

```
texto = "<table border=1><tr><th>Fruta</th><th>Cantidad</th></tr>";  
let frutas = mijson[0].Fruta;  
for (let i = 0; i < frutas.length; i++) {  
    fruta = frutas[i];  
    texto += "<tr><td>" + fruta.Nombre + "</td><td>" + fruta.Cantidad + "</td></tr>";  
}  
texto += "</table>";
```
- En caso de que haya elementos que puedan estar vacíos, debemos utilizar `try-catch` para evitar excepciones en Javascript.



# Formatos para envío y recepción de la información

## Ejercicio U7T4 – JSON:

Crea un archivo **JSON** equivalente al **XML** que creaste en el ejercicio anterior y comprueba su sintaxis (que está correctamente formado) con la herramienta online **JSON Viewer**.

Basándote en el ejemplo de la frutería, procesa el **JSON** de manera que obtengas el mismo resultado que obtuviste en el ejercicio anterior (la tabla de las series con las especificaciones dadas). Recuerda que solamente tendrás que modificar los siguientes archivos:

- **tvshows.json**: donde se encontrará la misma información del **XML** anterior pero en formato **JSON**.
- **index.js**: (o el nombre del fichero de script **JS** que hayas puesto) donde procesarás los datos teniendo en cuenta que se trata de un archivo **JSON**.

Será necesario utilizar una petición de tipo **GET** sobre la *url* directa al fichero **JSON**.



**END**



**prof.jduran@iesalixar.org**