

## Ejercicios JSON

1.- La estructura básica de un JSON consiste en pares clave-valor encerrados entre llaves {}. Cada clave se representa como una cadena de caracteres y se separa de su valor por dos puntos (:). Los pares clave-valor se separan entre sí por comas (,). El valor puede ser de diferentes tipos:

- Número (entero o de coma flotante)
- Cadena (entre comillas dobles)
- Booleano (verdadero o falso)
- Matriz (entre corchetes)
- Objeto (entre llaves)
- nulo

Crear una estructura JSON que contenga los siguientes tipos de datos JSON:

- Número, cadena, booleano, matriz (1x5), objeto, array (5 valores) y nulo. Los nombres de las claves en los pares clave-valor del JSON deben ser descriptivos del tipo de dato que contiene cada valor.

2.- Información de productos. Partiendo del siguiente fragmento de código XML que representa información de productos:

`<productos>`

`<producto>`

`<nombre>Camiseta</nombre>`

`<precio>19.99</precio>`

`<disponible>>true</disponible>`

`<detalles>`

`<color>Rojo</color>`

`<talla>M</talla>`

`</detalles>`

`</producto>`

`<producto>`

`<nombre>Pantalón</nombre>`

`<precio>29.99</precio>`

`<disponible>>false</disponible>`

`<detalles>`

`<color>Azul</color>`

`<talla>L</talla>`

`</detalles>`

`</producto>`

`</productos>`

Crea una estructura JSON que represente la misma información de los productos. La estructura JSON equivalente al fragmento de código XML se puede representar de varias formas, según si se eliminan o no las etiquetas raíz.

**3.- Sistema de gestión de inventario.** En un sistema de gestión de inventario, se manejan diferentes tipos de productos, cada uno con sus propias características. Cada producto tiene un código, nombre, precio y una categoría a la que pertenece. Además, se almacena información adicional como la fecha de ingreso y el stock disponible.

Crea un archivo JSON que represente esta información de los productos en el inventario. Utiliza la estructura de objetos y arrays JSON adecuada.

Escribe el código JSON resultante.

```
<inventario>
  <productos>
    <producto codigo="001" categoria="Electrónicos">
      <nombre>Televisor</nombre>
      <precio>599.99</precio>
      <fecha_ingreso>2023-07-15</fecha_ingreso>
      <stock>10</stock>
    </producto>
    <producto codigo="002" categoria="Ropa">
      <nombre>Camiseta</nombre>
      <precio>29.99</precio>
      <fecha_ingreso>2023-07-16</fecha_ingreso>
      <stock>50</stock>
    </producto>
    <producto codigo="003" categoria="Hogar">
      <nombre>Lámpara</nombre>
      <precio>39.99</precio>
      <fecha_ingreso>2023-07-17</fecha_ingreso>
      <stock>20</stock>
    </producto>
  <!-- Agrega más productos según sea necesario -->
</productos>
</inventario>
```

**4.-** Crea una página sencilla en la que se muestra el texto “demo”. Convierte un objeto JavaScript en un JSON string (cadena de caracteres JSON), y envíalo al servidor. Recuerda que la función `JSON.stringify()` permite tal conversión. Por otro lado, `window.location` nos permite enviar el JSON generado al servidor. En este ejemplo lo haremos mediante la ejecución de un php ficticio, al que llamaremos `demo_json.php`. Finalmente, almacena localmente el JSON generado, y a continuación cambia la propiedad del texto “demo” que aparece en la página por el valor de la propiedad “nombre” del objeto JS generado desde el JSON almacenado localmente.

El objeto que convertiremos en JSON es:

```
{nombre: "Mario", apellidos: "Sánchez López", edad: 26, curso: "DAW2", asignatura: "DEWC", ciudad: "El Puerto de Santa María"}
```

Nota: La sentencia que permite enviar al servidor es:  
`window.location = "demo_json.php?x=" + myJSON;`

**5.-** Convierte un JSON string en un objeto JavaScript. Recuerda que la función `JSON.parse()` permite tal conversión. El JSON string es:

```
{"nombre": "Mario", "apellidos": "Sánchez López", "nacido": "26-11-1986", "curso": "DAW2", "asignatura": "DEWC", "ciudad": "El Puerto de Santa María"}
```

En una página web sencilla en la que se muestra el texto “Datos del alumno”, sustituiremos el texto con la concatenación de los valores de todas las propiedades del objeto generado a partir del JSON string. La concatenación se puede hacer como la salida de un bucle `for` de los valores de las propiedades del objeto JS obtenido del JSON, o definiendo literalmente la concatenación de los valores de las propiedades del objeto.

Nota: Recuerda que el formato de fecha no es válido en JSON, es solo texto en JSON, por lo que después de pasar el JSON a objeto JS, deberás convertir la propiedad “nacido” a formato fecha con `“new Date()”`.