



7.A. Programación AJAX en JavaScript

2.- Envío y recepción de datos de forma asíncrona.

2.5.- Recepción de datos en formato XML.

Cuando realizamos una petición AJAX, que nos devuelve las respuestas en formato XML, dichos datos los tendremos que consultar en la propiedad `responseXML` del objeto XHR.

[Descarga el código fuente del ejemplo dwec0725.](#) (0.01 MB)

En la función `iniciar()`, le hemos dicho que cargue de forma asíncrona, empleando el método `GET`, el fichero `datosXML.php`. Esta aplicación PHP, nos devolverá un fichero XML, con una lista de `Cd` de música con el artista, país, compañía, etc.

Instrucción de carga del fichero `datosXML.php`: `cargarAsync("datosXML.php");`

Si queremos cargar directamente un fichero XML, y conocemos su nombre, podremos escribir directamente:

```
cargarAsync("catalogo.XML");
```

En la función de `estadoPetición()`, cuando `readyState` es 4 y el `status` es `OK` (200), accedemos a los resultados de la petición AJAX, en la propiedad `responseXML`. Para gestionar los datos XML, tendremos que recorrerlos empleando los métodos del DOM, ya que un fichero XML comparte la estructura en árbol de un documento HTML, y podemos utilizar, por tanto, los mismos métodos que empleamos para recorrer el DOM HTML.

En nuestro caso, lo primero que vamos a hacer es recorrer los elementos, que son los que contienen toda la información referente a los cd's de música:

```
// Almacenamos el fichero XML en la variable resultados.

resultados=this.responseXML;

// Tenemos que recorrer el fichero XML empleando los métodos del DOM

// Array que contiene todos los CD's del fichero XML

CDs= resultados.documentElement.getElementsByTagName("CD");
```

Haremos un bucle para recorrer todos los cd's del catálogo, y dentro de cada uno, imprimiremos los datos que nos interesen:

```
// Hacemos un bucle para recorrer todos los elementos CD.

for (i=0;i<CDs.length;i++)

{

... .
```

Dentro de cada CD, accederemos al elemento que nos interese e imprimiremos su contenido. Para imprimir el contenido de cada nodo, tendremos que hacerlo con el comando `try {} catch {}`, ya que si intentamos acceder a un nodo que no tenga contenido, nos dará un error de JavaScript, puesto que el elemento hijo no existe, y entonces se detendrá la ejecución de JavaScript y no imprimirá nuestro listado.

```
// Para cada CD leemos el título

titulos=CDs[i].getElementsByTagName("TITLE");

try { // Intentamos acceder al contenido de ese elemento
```



```

        salida+="<td>" + titulos[0].firstChild.nodeValue + "</td>";

    }

    catch (er) {      // En el caso de que no tenga contenido ese elemento imprimimos un espacio en
    blanco.

        salida+= " <td>&nbsp;</td>";

    }

```

Te proponemos echar un vistazo al siguiente código en el que se ha cambiado el paradigma de programación para que sea más funcional, de esta forma, como puedes comprobar, se reduce el tamaño del código y se hace más declarativo. Este cambio de paradigma puede costar asimilarlo al principio pero ofrece una visión del código menos imperativa y más acorde con las tendencias actuales de desarrollo de aplicaciones Web.

```

function estadoPeticion() {
    function td(elemento) {
        return `<td>${elemento.textContent}</td>`;
    }
    if (this.readyState==4 && this.status == 200) {
        // Almacenamos el fichero XML en la variable datos.
        var datos=this.responseXML;
        // Tenemos que recorrer el fichero XML empleando los métodos del DOM
        // Array que contiene todos los CD's del fichero XML
        var CDs=Array.from(datos.documentElement.getElementsByTagName("CD"));
        // En la variable salida compondremos el código HTML de la tabla a imprimir.
        salida="<table border='1'><tr><th>Titulo</th><th>Artista</th><th>Año</th></tr>";
        // Mapeamos una función que extrae los atributos de cada CD y compone las filas
        // de la tabla
        salida+=CDs.map(function(cd) {
            var atributos=cd.children;
            var arrcds=Array.concat(atributos[0],atributos[1],atributos[5]);
            return `<tr>${arrcds.map(td).join(" ")}`</tr>`;
        }).join(" ");
        // Cuando ya no hay más Cd's cerramos la tabla.
        salida+="</table>";
        // Desactivamos el indicador AJAX cuando termina la petición
        document.getElementById("indicador").innerHTML="";
        // Imprimimos la tabla dentro del contenedor resultados.
        document.getElementById("resultados").innerHTML=salida;
    }
}

```

Créditos de las imágenes

Autoría: Gideon Burton

Licencia: CC BY-SA 2.0.

◀ DWEC07 Solucion tarea

Ir a...

7.B. Introducción a AJAX en W3Schools ▶