

Ejercicios UT07

Los ejercicios se llevarán a cabo utilizando solo javascript y el objeto XMLHttpRequest.

Para la realización de algunos de los siguientes ejercicios con peticiones AJAX se necesitará un servidor web (XAMPP, por ejemplo)

Se deberán copiar los ficheros necesarios para cada ejercicio en una carpeta llamada **ut07**

El acceso a dichos ficheros será con la dirección **http://localhost/ut07/**

1. Crea un programa HTML con los siguientes elementos:

- Un div con identificador “demo”.
- Un botón llamado “Funcion Ajax”. De forma que cuando se pulse se muestre en el div con identificador “demo” el contenido del fichero de texto ej1UT7.txt.

El fichero de texto llamado ej1UT7.txt tendrá este contenido y estará guardado en el servidor en el mismo directorio que el programa HTML:

`
Esto es una prueba de Ajax
Espero que funcione
`

NOTA: Te puedes ayudar de este ejemplo:

https://www.w3schools.com/xml/tryit.asp?filename=tryajax_xmlhttp

2. Modifica el programa anterior para que cuando el estado de la petición XMLHttpRequest (propiedad readyState) muestre ese estado en el div “demo” siguiendo este esquema:

1. readyState = 1-> muestre el texto “Conexión establecida con el servidor”
2. readyState = 2 -> muestre el texto “El servidor ha recibido la petición”
3. readyState = 3 -> muestre el texto “El servidor está procesando la petición”,
4. readyState = 4 -> muestre el texto “Respuesta recibida del servidor”. Y evalúe si la respuesta es correcta, y sino lo es por qué.

3. Crea un programa HTML con los siguientes elementos:

- Un div con identificador “demo”.
- Un botón llamado “Calorias de las comidas del menú”. De forma que cuando se pulse se muestre en el div con identificador “demo” este mensaje para cada comida: “La comida XXXX tiene XX calorías”.

Para calcular las calorías de cada comida utilizará un fichero XML llamado ej3UT7.xml con este contenido:

```
<menu_almuerzo>
  <comida>
    <nombre>Waffles</nombre>
    <precio>$2.00</precio>
    <descripcion>Waffles baratos de McDonalds</descripcion>
    <calorias>650</calorias>
  </comida>
```

```

    <comida>
      <nombre>Hamburguesa</nombre>
      <precio>$5.00</precio>
      <descripcion>La hamburguesa mas comun de McDonalds</descripcion>
      <calorias>1500</calorias>
    </comida>
  </menu_almuerzo>

```

Este fichero XML se pedirá al servidor mediante AJAX.

NOTA: solo puedes utilizar `getElementsByTagName` y `childNodes`

4. Crea un programa HTML con los siguientes elementos:

1. Un `div` con identificador “demo”.
2. Un botón llamado “Temperatura y sensación térmica de los últimos días por franjas horarias”. De forma que cuando se pulse, se pida un fichero XML que facilitará a nuestro programa HTML estos datos. Ese fichero XML se pide a través de la URL: "https://api.openweathermap.org/data/2.5/forecast?q=Madrid,es&lang=es&units=metric&mode=xml&appid=479092b77bcf850403cb2aeb1a302425" con el método GET.

Lo que se tiene que mostrar por pantalla son frases como la siguiente para cada día y franja horaria facilitada en el fichero XML:

Dia 2021-02-26 desde las 18:00 hasta las 21:00 : Temperatura -> 13 Sensacion térmica -> 10.02

Visualiza previamente este fichero XML para poder conocer su estructura y así poder acceder a sus datos desde JavaScript.

5. Modifica el ejercicio anterior para que los datos los muestre en una tabla como ésta:

Dia	Hora inicio	Hora fin	Temperatura (°C)	Sensación térmica (°C)

Esta tabla se crea dinámicamente desde JavaScript al pulsar el botón “Temperatura y sensación térmica de los últimos días por franjas horarias”, no se crea en el fichero HTML. **IMPORTANTE:** Se debe crear utilizando los métodos de creación de nodos del arbol de nodos, no introduciendo directamente los códigos “<table>, <tr>, <td>....”

6. Crea un programa HTML con los siguientes elementos:

1. Un `div` con identificador “demo”.
2. Un formulario con:

1. Un input tipo texto para meter un Nombre de persona.

2. Un botón llamado “Comprobar nombre” de forma que cada vez que se haga click en él se ejecute un programa php. Este programa php funcionará de la siguiente forma:
 1. Se ejecuta con el método GET.
 2. Recibirá una variable llamada dato.
 3. El valor de esta variable será lo introducido por el usuario en el input tipo texto.
 4. Este programa php tendrá dentro un array con varios nombres de personas.
 5. Si el dato que le llega a través de la variable dato es igual a uno de estos nombres, el programa php devolverá OK, y si no es así devolverá NOK.

NOTA: El fichero PHP está disponible en el aula virtual.

3. Cuando se ejecute el programa php, si éste devuelve OK aparecerá en el div “Este nombre existe en el servidor”, y si devuelve NOK aparecerá en el div “Este nombre no existe en el servidor.”
-
7. Modifica la aplicación anterior de la siguiente manera:
 1. Crea un nuevo botón en el fichero HTML, llamado “Hacer algo”. Cuando se haga click en este botón se debe añadir al párrafo con id “demo”, la cadena “He hecho algo”.
 2. Modifica el fichero php añadiendo al mismo la sentencia sleep (10). Esta sentencia tarda en ejecutarse 10 segundos, por lo que la respuesta del fichero php va a tardar al menos 10 segundos.
 3. Ejecuta la aplicación y pulsa al botón “Comprobar nombre” (la respuesta tardará 10 segundos). Mientras tanto trata de hacer click al botón “Hacer algo”, ¿puedes?, ¿por qué?
 4. Modifica la aplicación para que la sentencia Ajax se ejecute de forma síncrona, y vuelve a hacer lo que hiciste en el apartado 3. ¿Qué ocurre?
 8. En este ejercicio vamos a obtener los datos de unos héroes de un fichero JSON. Este fichero JSON nos lo van a servir desde la URL: `'https://mdn.github.io/learning-area/javascript/ojs/json/superheroes.json'`

Para realizar la petición de este fichero JSON, lo vamos a hacer mediante Ajax, haciendo uso del objeto `XMLHttpRequest` (XHR) .

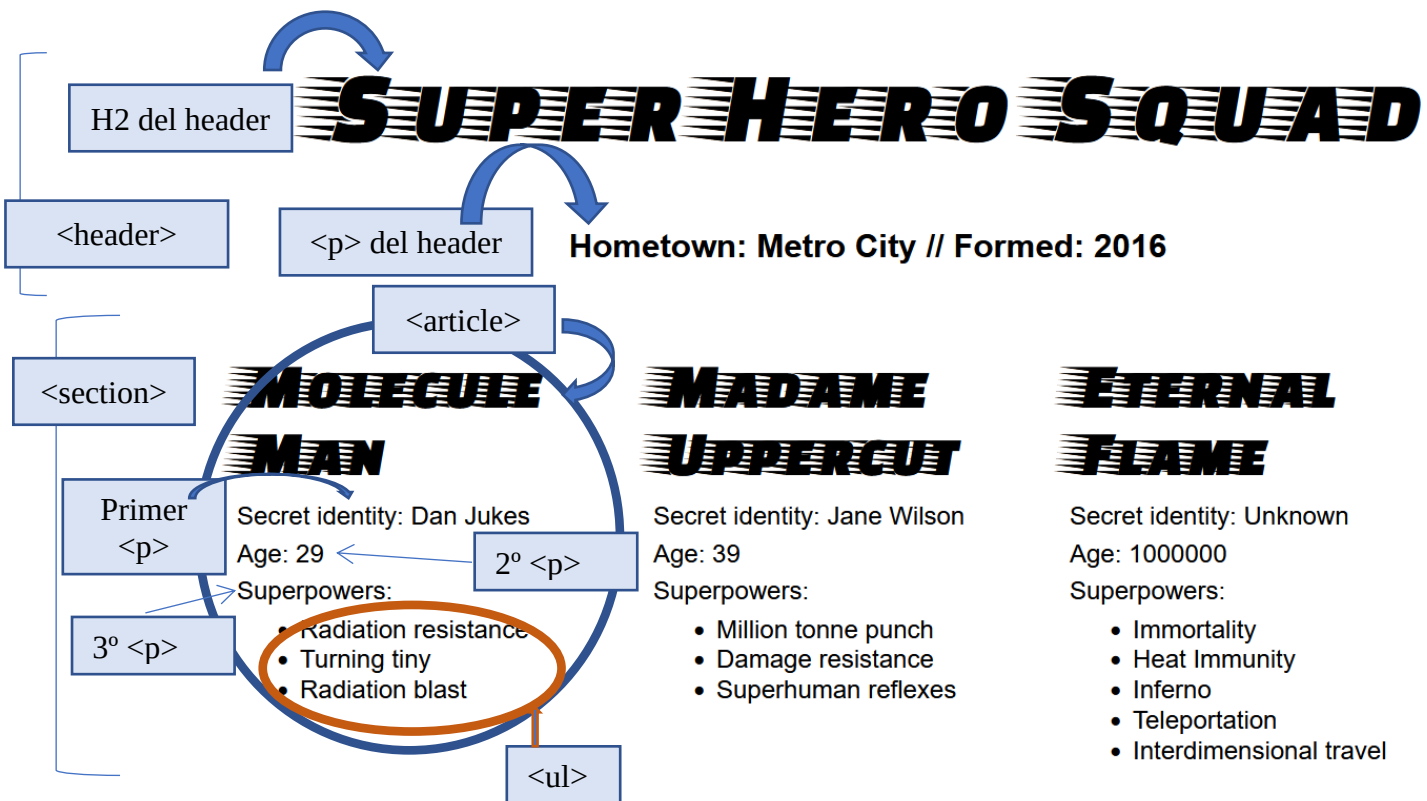
El contenido de este fichero JSON, lo vamos a recibir concretamente en la propiedad `responseText` del objeto XHR, es decir, en formato cadena.

Esta respuesta tipo cadena tenemos que pasarla a objeto JSON (que son objetos de JavaScript literales) mediante el método `JSON.parse(xhr.responseText)` . JSON es un objeto de JavaScript parecido a Math.

- Una vez recibido el fichero JSON que necesitamos vamos a mostrar los datos recibidos de la siguiente forma:
- Haremos uso del fichero `ej8UT7heroes.css` que hay publicado en el aula virtual. Este fichero no puede ser modificado.
- Haremos uso del fichero `ej8UT7heroes.html` que hay publicado en el aula virtual. Este fichero no puede ser modificado

- En la parte de JavaScript del fichero `ej8UT7heroes.js` se creará lo siguiente una vez recibidos los datos JSON:
 - En el `<body>` se creará lo siguiente:
 - un elemento HTML tipo `<header>`. Dentro de este `<header>` se crearán los siguientes elementos:
 - Un elemento `<h1>` que contendrá el dato `squadName` que obtendremos del fichero JSON.
 - Un elemento `<p>` que contendrá lo siguiente en una sola línea:
 - El literal `"Hometown: "`
 - El dato `homeTown` que obtendremos del fichero JSON.
 - El literal `"// Formed: "`
 - El dato `formed` que obtendremos del fichero JSON.
 - Un elemento HTML tipo `<section>`. Dentro de este elemento `<section>` crearemos tantos elementos HTML `<article>` como miembros de héroes haya. Cada `<article>` tendrá la siguiente estructura:
 - Un elemento `<h2>` que contendrá el dato `name` del superhéroe.
 - Un elemento `<p>` que contendrá lo siguiente:
 - El literal `"Secret identity: "`
 - El dato `secretIdentity` que obtendremos del JSON.
 - Otro segundo elemento `<p>` que contendrá lo siguiente:
 - El literal `"Age: "`
 - El dato `age` que obtendremos del JSON.
 - Un tercer elemento `<p>` que contendrá el literal `"Superpowers"`
 - Un elemento HTML lista ``, que contendrán tantos elementos ítems de la lista `` como superpoderes tengan el superhéroe. Cada `` de la lista contendrá el dato `Powers` que obtendremos del fichero JSON.

Un ejemplo de lo que se tiene que visualizar es lo siguiente (los estilos ya están programados en la hoja `ej8UT7heroes.css`)



9. Crea un formulario con los siguientes campos:

- Input tipo texto: Nombre.
- Input tipo texto: Apellidos.
- Input tipo texto para la provincia en la que vive.
- Un botón llamado “Ver datos introducidos”

Cada vez que el usuario teclee una tecla dentro del input tipo texto de la provincia, se va a hacer una consulta por Ajax a un fichero php pasándole el value de este input hasta el momento tecleado por el usuario. Este fichero php va a contener en un array todas las provincias de España, y va a devolver al cliente la lista de provincias que comienzan con los caracteres que hay introducidos en el input tipo text de la provincia del formulario

Al pulsar el botón “Ver datos introducidos” se va a visualizar todos los datos que el usuario ha introducido en el formulario.

El fichero PHP lo puedes construir ayudándote de este ejemplo, que es similar pero con nombres de personas: https://www.w3schools.com/js/js_ajax_php.asp

10. Crea un formulario con los siguientes campos:

- Input tipo texto: Nombre.
- Input tipo texto: Apellidos.
- Input tipo select: Comunidad autónoma en la que vive.
- Input tipo select: Provincia en la que vive.
- Un botón llamado “Ver datos introducidos”

El input tipo select para la provincia en la que vive el usuario va a funcionar de la siguiente manera:

- Las opciones se van a construir de forma dinámica en función de la Comunidad Autónoma que ha seleccionado el usuario.
- Las provincias de cada Comunidad Autónoma se van a consultar por Ajax a un fichero PHP cada vez que el select de la Comunidad Autónoma cambie. Este fichero PHP puede almacenar todas las provincias de todas las comunidades autónomas en arrays o cadenas, o como se quiera.

Al pulsar el botón “Ver datos introducidos” se va a visualizar todos los datos que el usuario ha introducido en el formulario.

Para resolver este ejercicio has de utilizar el fichero ej10UT7.php que está publicado en el aula virtual, que entregará la respuesta por Ajax al cliente a través de una cadena (xhr.responseText)

11. Repite el ejercicio 6, pero utilizando el método POST. Para ello necesitas saber lo siguiente:

- Se indica el tipo de método en open: `XHR.open('POST', "procesar.php", true);`
- Tenemos que crear una cabecera con el tipo de contenido que vamos a enviar, justo antes de enviar la petición con el método `.send()`:
`XHR.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");`
- En el método `.send()` escribiremos los parámetros que serán enviados por el método POST: `XHR.send(parámetros)`
- El fichero PHP no es necesario modificarlo si recoge el parámetro con `$_REQUEST`:
`$nombre = $_REQUEST["nombre"];`

12. Haz el ejercicio 11, pero utilizando el fichero ej12UT7.php que devolverá la respuesta al cliente en formato JSON. El fichero ej12UT7.php lo puedes encontrar en el aula virtual.

13. kjñlkjlñk