Unidad 1: Ajax y Callbacks

1. Introducción.
2. Lección 1: Que es un AJAX y como funciona.
   1. Que es AJAX y como funciona
   2. Los Callbacks dentro de AJAX

La importancia de los callbacks en AJAX



En cursos anteriores aprendimos qué son los callbacks y cómo estos nos hacen la vida más fácil a la hora de programar usando el lenguaje JavaScript.  
Aunque en Javascript lo más común es utilizar librerías o paquetes de terceros para realizar la técnica AJAX, en este curso vamos aprender a hacerlo de las dos maneras, usando Javascript puro y luego usando librerías como jQuery. Los estados por los que pasa una petición XMLHttpRequest (Javascript puro) son los siguientes:

* Petición no inicializada
* Conexión establecida con el servidor
* Petición recibida por el servidor
* La petición está siendo procesada
* La petición finalizó y la respuesta está lista

Con Javascript puro es fácil detectar el estado por el que está pasando la petición pero cuando estamos usando librerías como jQuery lo único que podemos hacer para controlar el tiempo en el que la petición esta lista es usando un callback, por eso para continuar con las unidades siguientes debes tener claro este concepto.

Ejemplo de AJAX usando jQuery y empleando 2 callbacks:



* 1. Ejemplos de AJAX Básico.

$.ajax({

url: ‘algunaURL/algo’,

type: ‘GET’, // Puede ser POST,PUT

data: {},

success: function(respuesta){ 🡪 Estado numero 4 de la funcion

// Hacer algo con la respuesta

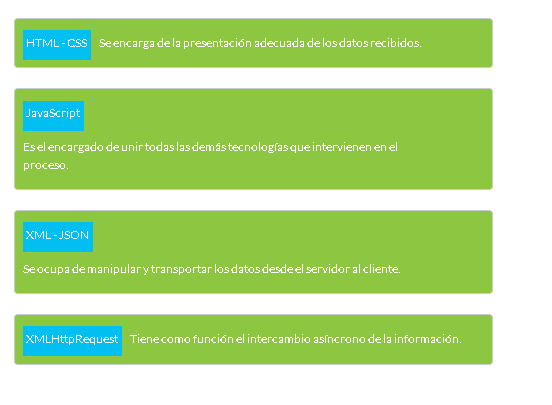
}

})

url: <http://api.tvmaze.com/shows>

* 1. Actividad.

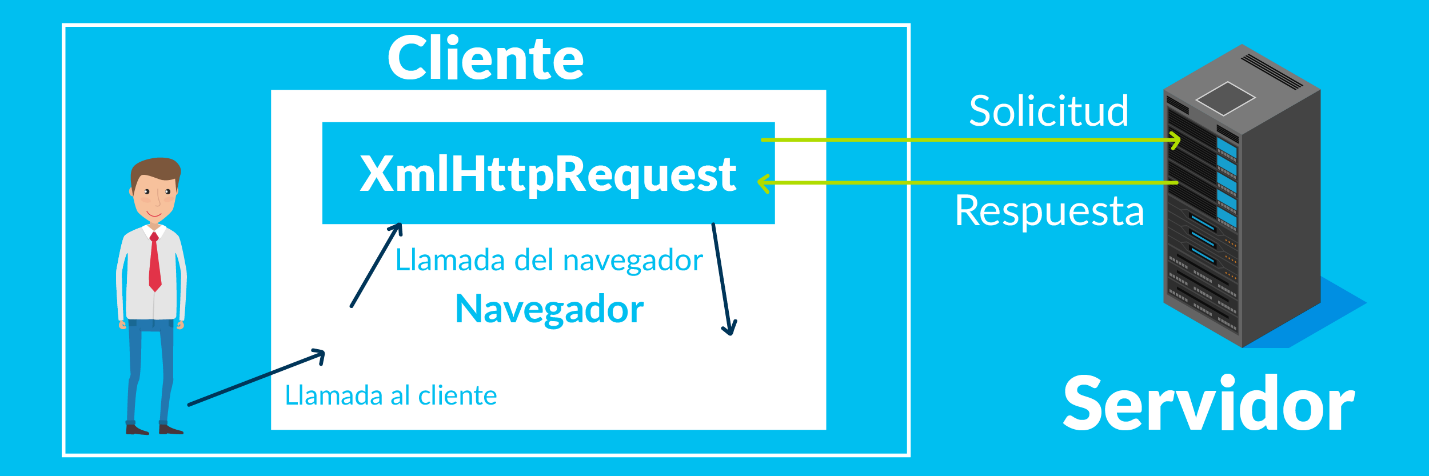
En los recursos de esta lección conociste un nuevo concepto: Ajax.  
Ésta es una de las tecnologías más usadas en la actualidad ya que permite la comunicación entre varios sistemas para actualizar las páginas web dinámicamente sin necesidad de actualizar la página en el navegador. Su nombre viene de “Asynchronous Javascript and XML” y, aunque conserva su nombre, la tecnología XML ya no es tan popular y por el contrario se usa JSON.  
  
Instrucciones:  
  
En las siguientes columnas se encuentran: en un costado algunas tecnologías que hacen parte o intervienen en Ajax y el en otro la función que cumplen dentro del proceso de peticiones asíncronas al servidor. Relaciona el concepto de la columna izquierda con su correspondiente función de la columna derecha.



1. Lección 2: Introducción al XMLHttpRequest.
   1. Qué es XMLHttpRequest

XMLHttpRequest (XHR)

XMLHttpRequest o XHR como es conocido en el mundo del desarrollo web, es un objeto JavaScript desarrollado por Microsoft y adoptado por Google, Apple y Mozilla. Este objeto se convirtió en un estándar mantenido por la W3C y proporciona una funcionalidad capaz de obtener información de una manera fácil de una URL específica, sin necesidad recargar la página. Mediante este proceso podemos recargar solo una parte de la página sin bloquear lo que el usuario se encuentra haciendo.

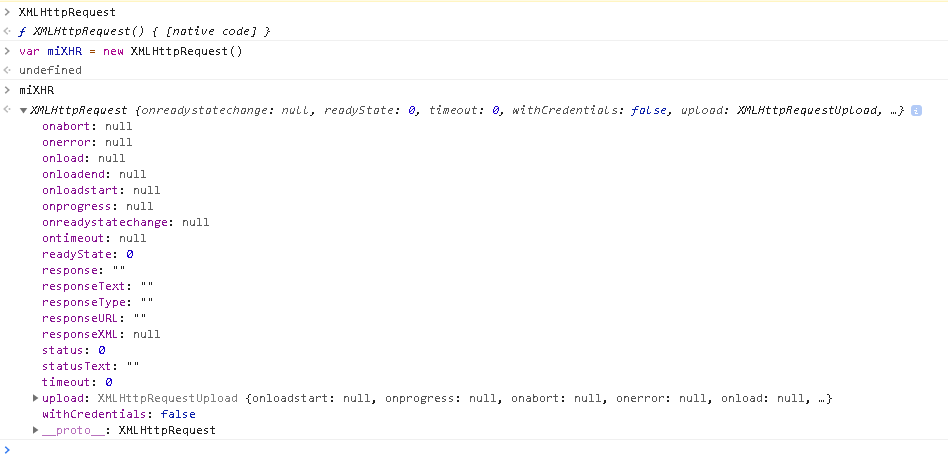


XMLHttpRequest tiene un conjunto de métodos y propiedades que veremos en el siguiente tema y a pesar de su nombre, XML-HttpRequest, este objeto permite transferir cualquier tipo de dato, no necesariamente debe ser XML.

* 1. El objeto XMLHttpRequest, sus métodos y propiedades

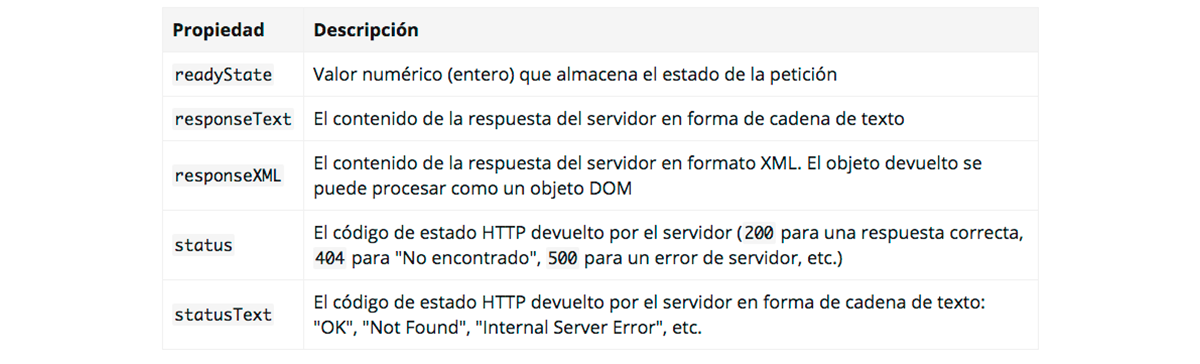
En esta ocasión veremos cómo está compuesto este objeto y cuáles son los métodos y propiedades que se encargan de realizar la funcionalidad.  
  
Si abrimos una consola en el navegador, escribimos XMLHttpRequest y pulsamos Enter, podremos ver que es una función que corresponde a código nativo JavaScript:  
  
  


Para poder ver su contenido, los métodos y las propiedades, es necesario instanciar un nuevo objeto a partir de esta función, el cual heredará todo su contenido. Esto lo hacemos de la siguiente manera:

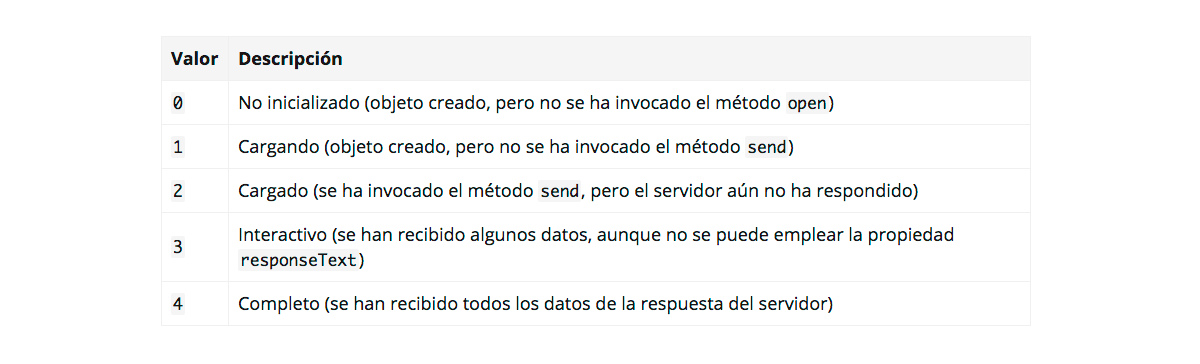


Resumen de los métodos y propiedades

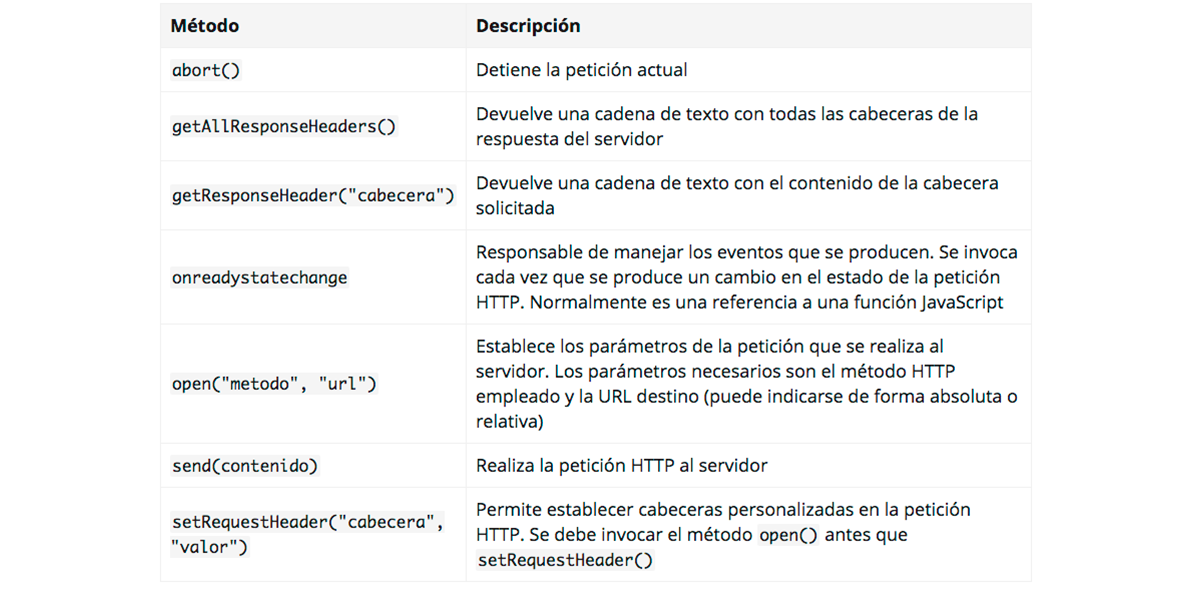
Propiedades



Valores de la propiedad readyState

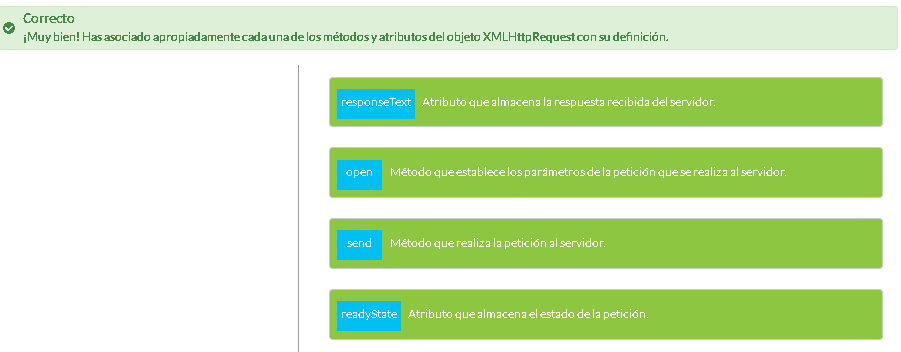


Métodos



* 1. Ejemplo de XMLHttpRequest – Done
  2. Actividad

En en esta lección conocimos el mundo de Ajax específicamente exploramos el objeto XMLHttpRequest. Éste permite realizar las peticiones de Ajax al servidor y gestionar las respuestas, los errores y demás procesos relacionados. Sin embargo dicho objeto posee una serie de métodos y atributos que debes dominar para sacar el mayor provecho a Ajax.  
  
Instrucciones:  
  
A continuación verás en la columna de la izquierda algunos métodos o atributos del objeto XMLHttpRequest y en la columna de la izquierda la función que cumplen cada uno de ellos. Relaciona cada elemento de la izquierda con su respectiva explicación en la derecha.



1. Lección 3: Ejecución sincrónica y asincrónica.

4.1 Peticiones Sincrónicas

XMLHttpRequest = asíncrono por defecto.

XHR.open('GET',URLUser, false) 🡪 Con false hacemos la solicitud síncrona

4.2 Ventajas y Desventajas de AJAX

Recordemos las tecnologías que hacen parte de AJAX



Ventajas

* Minimiza el número de transferencias de datos entre cliente y servidor.
* Optimiza los recursos y el tiempo de las operaciones.
* Mejora la experiencia de usuario al permitir traer datos desde el servidor, sin recargar la página.
* Puede ser usado en cualquier navegador web que tenga activo JavaScript.

Desventajas

* Utilizar AJAX para todo, incluidos formularios recuperación de contraseña.
* Los tiempos de respuesta del servidor pueden ser cruciales en determinadas peticiones que manejan grandes volúmenes de datos.
* Es necesario tener conocimiento sobre todas las tecnologías que hacen parte de AJAX para lograr usarlo de una manera correcta.

4.3 Ejemplos de peticiones sincrónicas y asincrónicas usando JQuery

Metodo AJAX

4.4 Actividad

 Tiempo de pensar

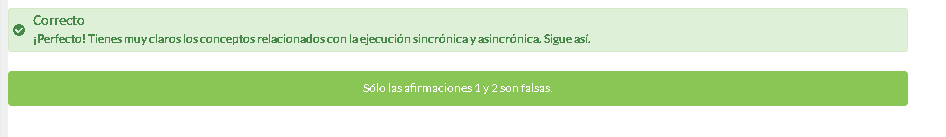
En los recursos de esta lección aprendiste sobre la ejecución sincrónica de las peticiones que, a diferencia de la ejecución asincrónica convencional, evita que el navegador realice otras operaciones mientras se espera la respuesta del servidor.  
  
Instrucciones:

Lee las afirmaciones que se muestran a continuación y determina si su contenido es verdadero o falso en relación con Ajax.

Luego elige la opción correcta.

Afirmaciones:

* El método de envío de peticiones sincrónico otorga eficiencia y fluidez a una página web.
* La ejecución asincrónica será descontinuada en poco tiempo.
* Tanto la ejecución sincrónica como la asincrónica pueden usarse en cualquier navegador que soporte JavaScript.



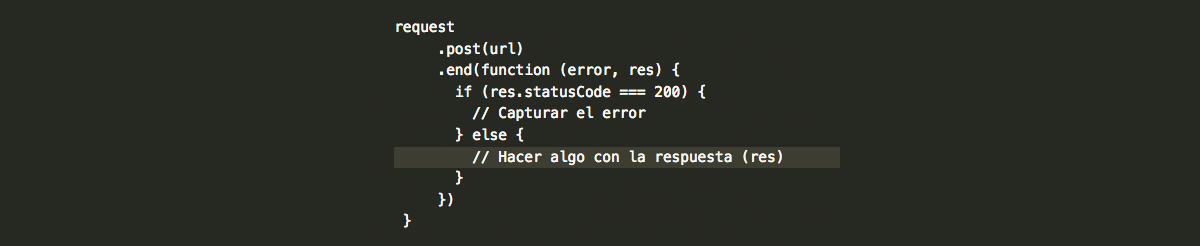
1. Lección 4: Utilidades y Objetos AJAX.

5.1 Diferentes librerías para AJAX y cómo funcionan

Diferentes librerías para AJAX y cómo funcionan

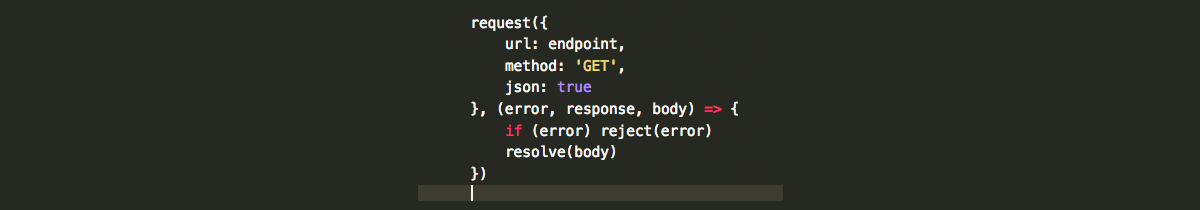
En la actualidad existen muchas librerías que facilitan la forma cómo realizamos peticiones AJAX a un servidor de datos. La más popular, sin duda, es jQuery y su método $.ajax. Pero existen alternativas, incluso dentro del mismo jQuery, como $.get o $.post.  
Estas son algunas de las alternativas a los métodos de jQuery:

superagent

Pequeña librería construida en Javascript, que utiliza el objeto XMLHttpRequest pero nos provee una forma más intuitiva y legible de realizar peticiones AJAX:  
Ejemplo:  
  
Repositorio de GitHub: <https://github.com/visionmedia/superagent>  
Documentación: <http://visionmedia.github.io/superagent/>

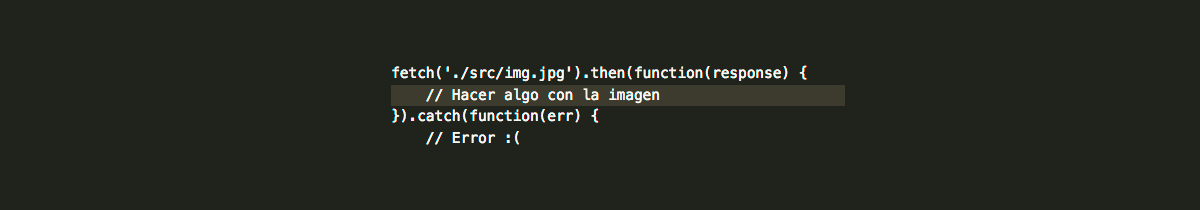
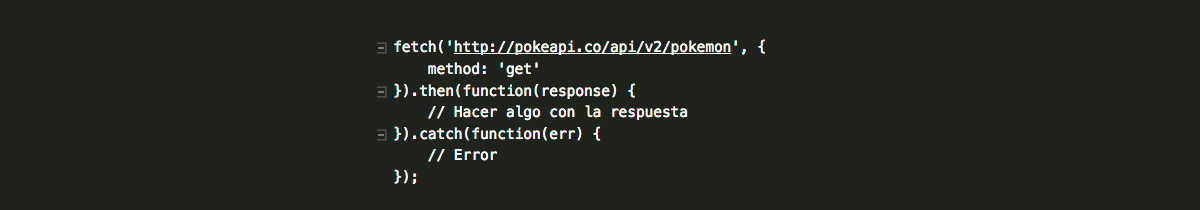
request

Es otra de las alternativas que nos sirve para realizar todo tipo de peticiones http de tipo AJAX, incluidas GET, POST, PUT, DELETE y que está escrita en Javascript. Esta librería también utiliza el objeto XMLHttpRequest y permite realizar, de una forma más elegante, este tipo de operaciones.

Ejemplo:  
  
Repositorio de GitHub: <https://github.com/request/request>

Fetch API

API que está incluida en JavaScript por defecto, o sea que no es una librería sino una función que hace parte del core del lenguaje. La función fetch () está disponible en el scope global por lo que puede ser accedida desde cualquier lugar de nuestra aplicación.  
Fetch está hecha para obtener recursos, incluso datos que se encuentran en un servidor externo. Así, la API no utiliza el objeto XMLHttpRequest sino una personalizada interfaz construida en base a los objetos Request y Response propios del protocolo HTTP.

Ejemplo: Recurso común  
  
Ejemplo HTTP  


5.2 jQuery.ajax()

Ejemplo:

for (var i=1;i<=80;i++){

$.ajax({

url: 'http://pokeapi.co/api/v2/pokemon-form/' + i,

type: 'GET',

data: {},

success: function(data){

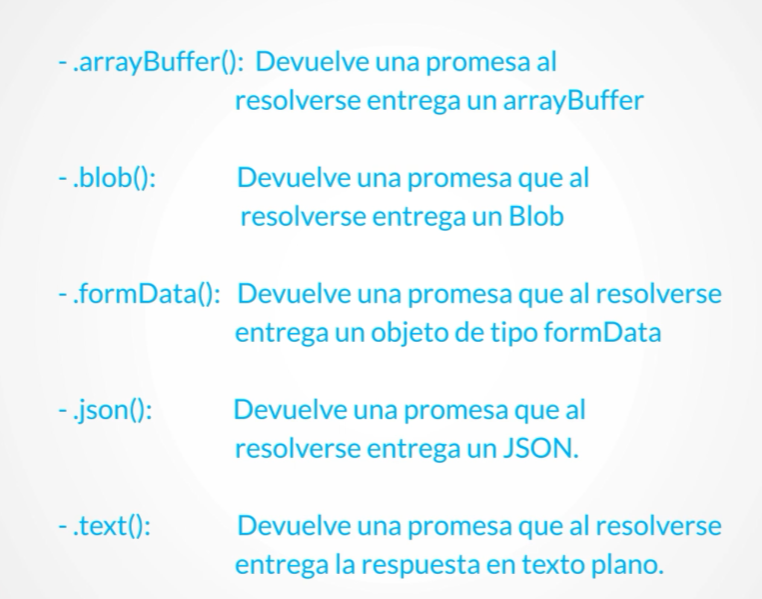
$('.pokemons').append('<li><img src="" ></img>' + data.name + '</li>')

}

})

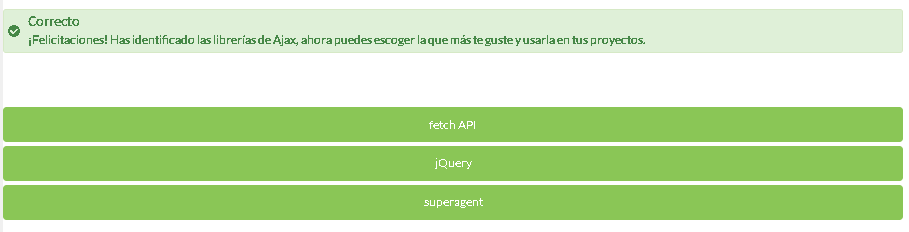
}

5.3 Fetch



5.4 Actividad

En los recursos de esta lección tuviste la oportunidad de conocer métodos alternativos al XMLHttpRequest para realizar peticiones Ajax. El uso de diversas librerías que pueden ser usadas de forma libre, hacen de tus proyectos más fáciles de codificar en cada petición que debas realizar a un servidor o una API en particular.  
  
Instrucciones:  
  
Lee la pregunta que se plantea a continuación y selecciona tres respuestas correctas.  
  
Pregunta:  
  
De las siguientes librerías, ¿Cuáles se encargan de realizar peticiones Ajax de una manera simplificada?



5.5 Ejercicio de Codificación

En las redes sociales o en los medios donde se publicán artículo, siempre existe un área de comentarios acerca del contenido que se provee.

En este ejercicio queremos simular la carga de comentarios relacionados a un artículo. La carga será vía AJAX ya que no se deben mostrar todos los comentarios debido a que pueden ser muchos y el contenido más reciente es el que usualmente importa.

La carga de comentarios se puede hacer de dos maneras:

Presionando el botón Ver más

Haciendo scroll hacia abajo en toda lá página, Este último comportamiento es común en muchos sitios donde no es necesario presionar un botón para seguir viendo el contenido.

Cada comentario tiene un contenido, el autor y la fecha con que se realizó.



El contenido a cargar será siempre el mismo y serán dos comentarios incluidos en un JSON al cual se le llamara de manera asíncrona.

Para que las llamadas AJAX funcionen el código debe servirse con un servidor web para que el navegador puede ejecutarlo sin problema.

En este caso te recomendamos instalar un servidor web en tu computador personal. La forma más sencilla de hacer esto de manera universal (MacOSX, Windows y Linux) es instalando el lenguaje Python que trae consigo un pequeño y simple servidor web. Para correr este servidor web solo debes ejecutar python -m SimpleHTTPServer 8000 desde la terminal de comandos para en el directorio donde están tus archivos y luego abres tu navegador con la direccion a <http://localhost:8000/index.html>

Notas:

python3 -m http.server 7800

python -m SimpleHTTPServer 7800

Instrucciones

1. Descarga el [archivo base](https://s3.amazonaws.com/nextu-content-production/Desarrollador_Web/06_BackEnd/Ejercicios/WEB16S_C6U1L4_Ejercicio1/base.zip).
2. Agrega al menos dos comentarios con los datos necesarios (autor, fecha y contenido).
3. Completa los estilos para que la interfaz luzca como comentarios.
4. Crea una función que cargue vía **AJAX** usando **JQuery** los comentarios del archivo *comentarios.json*
5. Provee el mismo comportamiento al hacer el *scroll* y llegar al final de la página.
6. Crea una función que cargue vía **AJAX** los comentarios pero esta vez usando **fetch**

Buenas Prácticas

* Al crear la función *cargarMasComentarios()* puede ser llamada desde distintos activadores.
* Toma en cuenta que la funcion *fetch* no es soportada por todos los navegadores.
* En algunos casos es importante poder manejar el caso en el que la llamada no es exitosa.

SOLUCIÓN

Asegúrate de realizar la actividad y hallar tu propia solución.

También encontrarás una propuesta que te ofrecemos como guía para reforzar tus conocimientos. Ejercitándote podrás mejorar tus habilidades para resolver este tipo de problemas.

Si se te dificulta algún paso de la actividad recuerda que **tu tutor en línea**se encuentra disponible para ayudarte y resolver cualquier duda o inquietud que tengas. Es importante que aprendas estos conceptos para tu formación como Desarrollador Web. Así que utiliza el archivo de solución únicamente para verificar tu trabajo y asegurarte de que vas por el camino correcto. ¡Éxitos!