Mètodes i propietats de l'objecte XMLHttpRequest

L'objecte XMLHttpRequest té moltes altres propietats i mètodes diferents a les que es feien servir per la primera aplicació d'AJAX.

Propietat	Descripció
onreadystatechange	Defineix la funció a ser cridada quan la propietat readyState canvia
readyState	Conté l'estat de l'objecte XMLHttpRequest ✓ 0: petició no inicialitzada ✓ 1: petició no inicialitzada ✓ 2: petició no inicialitzada ✓ 3: petició no inicialitzada ✓ 4: petició no inicialitzada
responseText	Retorna el contingut de la resposta del servidor en forma de cadena de text
responseXML	Retorna el contingut de la resposta del servidor en forma de cadena de text. L'objecte retornat pot ser processat com un objecte DOM
status	Retorna el codi d'estat HTTP retornat pel servidor ✓ 1xx: Informació. ✓ 2xx: Exitós. ✓ 3xx: Redirecció. ✓ 4xx: Error del Client. ✓ 5xx: Error del Servidor. ✓ Per obtenir la llista sencera visitar la pàgina https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp
statusText	Retorna el text associat al codi d'estat HTTP retornat pel servidor (i.e. "OK", "Not Found", etc)

Mètode	Descripció
abort()	Cancel·la la sol·licitud actual
getAllResponseHeaders()	Retorna una cadena de text amb totes les capçaleres de la resposta del servidor
getResponseHeader(capçalera)	Retorna la informació específica de la capçalera sol·licitada
open(metode, url, async, usuari, password)	Especifica el tipus de sol·licitud, la URL, si la sol·licitud s'ha de gestionar de manera asíncrona o no, i tots els atributs opcionals de la sol·licitud ✓ mètode: indiquem el tipus de sol·licitud (GET o POST) ✓ url: l'adreça del fitxer al que enviem les peticions en el servidor ✓ async: true(asíncrona) o false (síncrona) - opcional ✓ usuari i contrasenya: si fos necessària l'autenticació al servidor - opcional
send (dades)	Envia peticions HTTP al servidor ✓ dades: es fa servir en el cas en què estiguem utilitzant el mètode POST com a mètode d'enviament. Si fem servir el mètode GET, dades serà null.
setRequestHeader(capçalera, valor)	Permet establir capçaleres personalitzades en la sol·licitud HTTP. Afegeix el parell etiqueta/valor a la capçalera que s'enviarà al servidor. Cal invocar el mètode open() abans que setRequestHeader()
onreadystatechange()	És el responsable de manegar els events que es produeixen. Aquest mètode s'invoca cada vegada que es produeix un canvi en l'estat de la sol·licitud HTTP. Generalment és una referència a una funció JavaScript.

Com hem comentat anteriorment, per defecte les peticions es fan de manera asíncrona (tercer paràmetre del mètode **open**). Si indiquem un valor **false**, la petició es realitza de manera síncrona, és a dir, l'execució de l'aplicació s'atura fins que es rep la resposta completa del servidor. Però pel que hem pogut veure fins ara, fer les peticions síncrones va en contra totalment de la filosofia d'AJAX ja que aquest tipus de peticions *congelen* el navegador i no permet cap mena d'interacció de l'usuari amb el navegador. La sensació que rep l'usuari és que el navegador s'ha *penjat* amb la qual cosa no és recomanable fer peticions síncrones a no ser que sigui estrictament necessari.

D'altra banda, el mètode **send()** requereix d'un paràmetre que indica la informació que es vol enviar al servidor juntament amb la petició HTTP. En cas que no vulguem enviar dades indicarem **null**. Si al contrari, fem la petició amb el mètode **POST** i hem d'enviar dades, podem enviar com a paràmetres una cadena de text, un array de bytes o un objecte XML DOM.

Activitat 2

La pàgina HTML proporcionada inclou una zona anomenada ticker en què s'han de mostrar notícies generades pel servidor. Afegir el codi JavaScript necessari per:

1. Periòdicament cada cert temps (per exemple cada 5 segon) es realitza una petició al servidor mitjançant AJAX i es mostra el contingut de la resposta en la zona reservada per a les notícies.
2. A més del contingut enviat pel servidor, s'ha de mostrar l'hora en què s'ha rebut la resposta.
3. Quan es polsi el botó "Aturar", l'aplicació atura les peticions periòdiques al servidor. Si es torna a polsar sobre aquell botó, es tornen a realitzar les peticions periòdiques.
4. Afegir la lògica dels botons "Anterior" y "Següent", que aturen les peticions al servidor i permeten mostrar els continguts anteriors o posteriors al que es mostra en aquell moment.
5. Quan se rep una resposta del servidor, es ressalta visualment la zona anomenada ticker.
6. Modificar l'aplicació per a que es reutilitzi contínuament el mateix objecte XMLHttpRequest per fer les diferents peticions.