

EXERCICIS UNITAT DIDÀCTICA 8

EXERCICI 8.1

Es proporciona una pàgina web que té un <div> anomenat info.

S'ha d'implementar el codi necessari perquè es pugui accedir de manera asíncrona a una pàgina de text anomenada "Exercici 8.1.txt" i mostrem el seu contingut en el div info.

Exercici 8.1.txt y la solució de l'exercici han d'estar en el directori htdocs del nostre XAMPP

EXERCICI 8.2

Es proporciona una pàgina web.

a) Al carregar la pàgina, la caixa de text ha de mostrar per defecte l'URL de la mateixa pàgina.

b) Al pulsar el botó "Mostrar Contenidos", s'ha de descarregar per mitjà de peticions AJAX el contingut corresponent a la URL introduïda per l'usuari. El contingut de la resposta rebuda del servidor s'ha de mostrar en la zona de "Contenidos de l'archivo".

c) En la zona "Estados de la petición" s'ha de mostrar en tot moment l'estat en què es troba la petició (No inicialitzada, carregant, completada, etc.)

d) Mostrar el contingut de totes les capçaleres de la resposta del servidor en la zona "Cabeceras HTTP de la respuesta del servidor".

e) Mostrar el codi i text d'estat de la resposta del servidor en la zona "Código de estado".

Nota : El fitxer de la solució s'ha de pujar al servidor XAMPP

EXERCICI 8.3

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

- a) De forma periòdica cada un cert temps (per exemple cada 5 segons) es realitza una petició al servidor per mitjà d'AJAX i es mostra el contingut de la resposta en la zona reservada per a les notícies.
- b) A més a més del contingut enviat pel servidor, s'ha de mostrar l'hora en què s'ha rebut la resposta.
- c) Quan es polse el botó “Detener”, l'aplicació deté les peticions periòdiques al servidor. Si es torna a pulsar sobre eixe botó, es reprenen les peticions periòdiques.
- d) Afegir la lògica dels botons “Anterior” i “Siguiente”, que detenen les peticions al servidor i permeten mostrar els continguts anteriors o posteriors a què es mostra en eixe moment.
- e) Quan es rep una resposta del servidor, es ressaltava visualment la zona crida ticker.
- f) S'ha de utilitzar contínuament el mateix objecte XMLHttpRequest per a fer les diferents peticions.

Nota : Tant el fitxer “Exercici 8.3.php” com la solució han d'estar en el servidor XAMPP

EXERCICI 8.4

Un exemple de validació complexa és la que consistix a comprovar si un nom d'usuari triat està lliure o ja ho utilitza un altre usuari.

Com és una validació que requerix l'ús d'una base de dades molt gran, no es pot realitzar en el navegador del client.

Utilitzant les tècniques mostrades anteriorment i la pàgina web que es proporciona:

1. Crear un script que comprove amb AJAXy l'ajuda del servidor si el nom triat per l'usuari està lliure o no.
2. El script del servidor s'anomena “Exercici 8.4.php” i el paràmetre que conté el nom s'anomena login.
3. La resposta del servidor és “si” o “no”, en funció de si el nom d'usuari està lliure i es pot utilitzar o ja ha sigut ocupat per un altre usuari.
4. A partir de la resposta del servidor, mostrar un missatge a l'usuari indicant el resultat de la comprovació.

Nota : Tant el fitxer “Exercici 8.4.php” com la solució han d'estar en el servidor XAMPP

EXERCICI 8.5

Normalment, quan es valida la disponibilitat d'un nom d'usuari, es mostra una llista de valors alternatius en el cas que el nom triat no estiga disponible.

a) Modificar l'exercici de comprovació de disponibilitat dels noms perquè permeti mostrar una sèrie de valors alternatius tornats pel servidor.

Fer una versió utilitzant la forma tradicional (versió 1) i una altra versió utilitzant la funció fetch () (versió 2)

El script del servidor s'anomena "Exercici 8.5-a.php" i el paràmetre que conté el nom s'anomena login .

La resposta del servidor és un document XML amb l'estructura següent :

Si el nom d'usuari està lliure :

```
<respuesta>
<disponible>si</disponible>
</respuesta>
```

Si el nom d'usuari està ocupat :

```
<respuesta>
<disponible>no</disponible>
<alternativas>
<login>...</login>
<login>...</login>
...
<login>...</login>
</alternativas>
</respuesta>
```

Els noms d'usuari alternatius s'han de mostrar en forma de llista d'elements ().

b) Realitzar el mateix exercici però ara la informació es rep en un JSON amb la mateixa estructura que en l'apartat a)

Fer una versió utilitzant la forma tradicional (versió 1) i una altra versió utilitzant la funció fetch () (versió 2)

El script del servidor s'anomena ara "Exercici 8.5-b.php" i el paràmetre que conté el nom s'anomena login .