



UD7: Comunicación asíncrona con AJAX -----RESUMEN-----

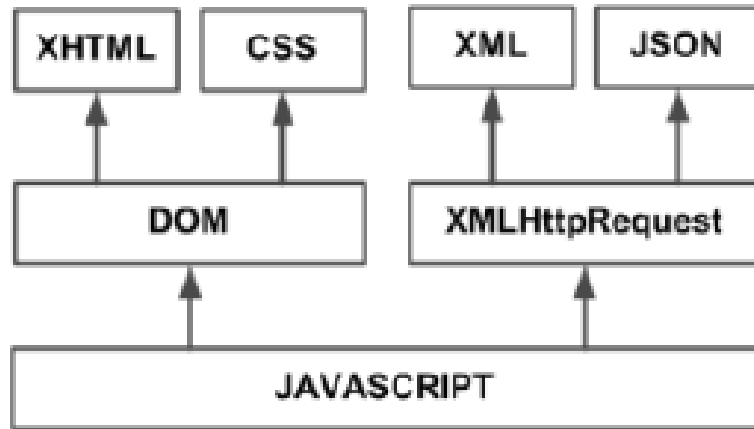
Antonio Sierra

Basado en el trabajo de Javier G. Pisano

AJAX

(Asynchronous Javascript and XML)

- Es una tecnología que nos va a permitir comunicar cliente con servidor en un segundo plano (sin tener que esperar por la respuesta)



Será necesario tener un servidor Web al que haremos las peticiones



El objeto XMLHttpRequest

UD7: Comunicación asíncrona con AJAX

Objeto XMLHttpRequest

- ▶ Objeto integrado en la especificación ECMAScript.
- ▶ Se utiliza para realizar y procesar las peticiones HTTP directamente al servidor Web
 - ▶ Es un “intermediario” entre el cliente y el servidor

Objeto XMLHttpRequest

Características

- ▶ Permite hacer tanto llamadas **síncronas** (espera por respuesta) como **asíncronas** (no espero por respuesta), si bien lo habitual es lo segundo.
- ▶ Por **seguridad**, solo deja hacer llamadas a páginas que se encuentran hospedadas en el mismo dominio desde el cual se realiza la petición AJAX

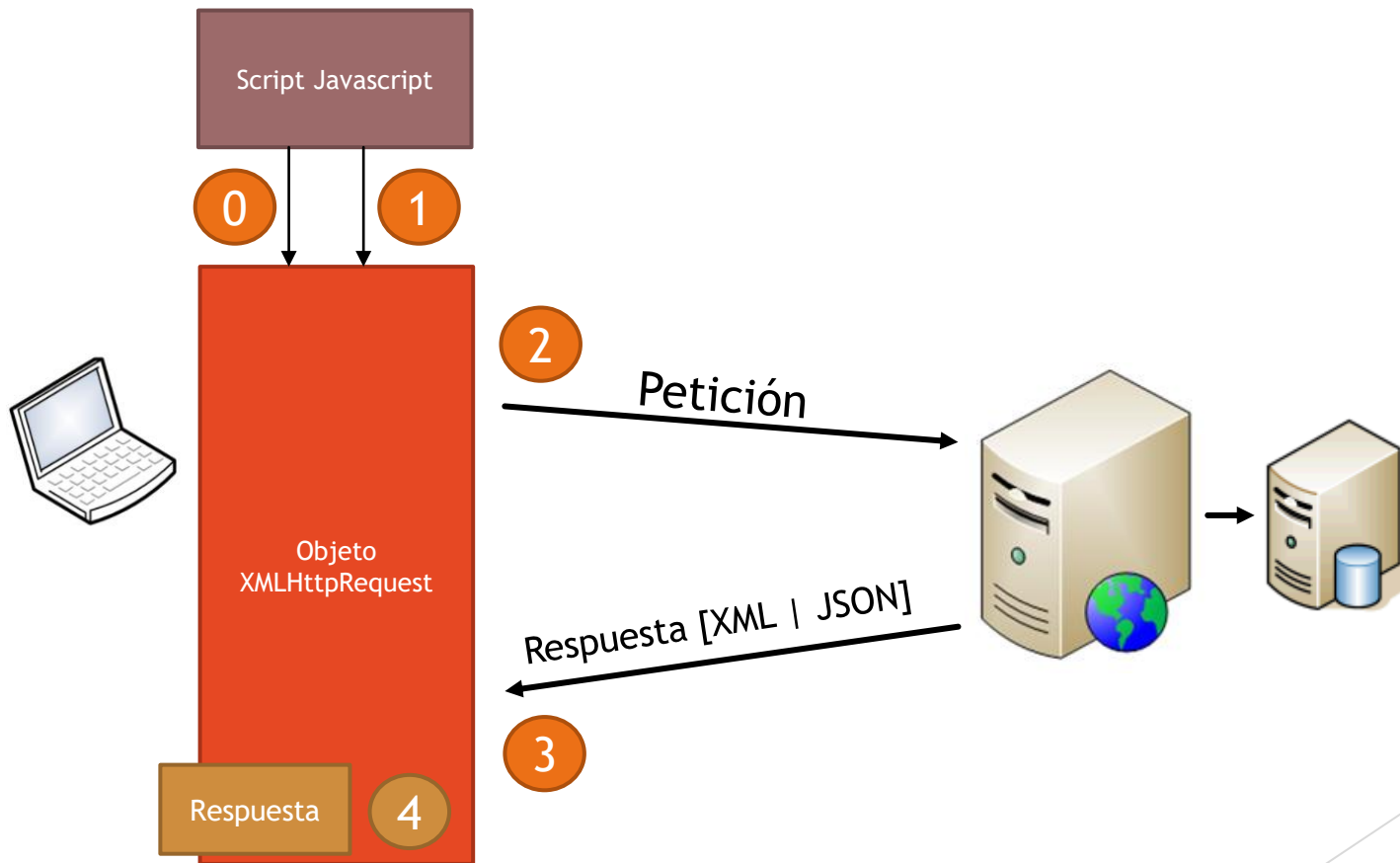
Objeto XMLHttpRequest

Propiedades

readyState	Valor numérico que almacena el estado de la petición
responseText	Contenido de la respuesta del servidor en formato de texto
responseXML	Contenido de la respuesta del servidor en formato XML
status	Código de estado HTTP devuelto por el servidor
statusText	Código de estado HTTP devuelto por el servidor [cadena]

200: OK
404: No encontrado
500: Error de servidor

Valores de la propiedad readyState



Pasos en una petición AJAX

- ▶ Instanciar el objeto XMLHttpRequest
- ▶ Preparar la función de respuesta
 - ▶ Escribimos la función de callback que asignamos a la propiedad **onreadystatechange**
- ▶ Realizar la petición al servidor y/o enviamos parámetros
- ▶ Se ejecuta la función de respuesta
 - ▶ Comprobamos si se ha recibido respuesta (**readyState** es 4) y si es correcta (**status** es 200).
 - ▶ Una vez realizadas las comprobaciones, accedemos a la respuesta del servidor mediante **responseText** ó **responseXML**

Procesando respuestas

- ▶ Si bien las respuestas pueden ir en texto plano, lo habitual es que las respuestas del servidor vayan en uno de los siguientes formatos:
 - ▶ XML
 - ▶ Tecnología soportada por la mayor parte de lenguajes y madura (usada en un gran número de aplicaciones).
 - ▶ JSON
 - ▶ Más compacto y ligero
 - ▶ Más fácil de procesar en el navegador.

Evitando el cacheado

- ▶ Un problema asociado a las peticiones AJAX puede ser la caché del navegador
 - ▶ Si repetimos las peticiones al mismo script y con los mismos parámetros, el navegador puede cachear el resultado
- ▶ Posibles soluciones:
 - ▶ Especificar en la cabecera HTTP (desde PHP) que no queremos cachear el resultado.
 - ▶ No es fiable al 100%
 - ▶ Añadir un parámetro aleatorio (llamado nocache) a todas las peticiones
 - ▶ Así cada petición varía en el valor de al menos uno de los parámetros.