

## Tarea 8.1 Mi primera comunicación AJAX

Utilizando Ajax, realiza una comunicación con el servidor que reciba un fichero texto. El funcionamiento será el siguiente:

- Página html con un botón que cargue en un div el fichero denominado "hola.txt"
- El fichero "hola.txt" contendrá un respetuoso saludo y presentación con tu nombre, del tipo: "Bienvenido, soy Fulanito de Copas y estoy a su entera disposición"
- Asegúrate de mantener la compatibilidad con navegadores antiguos.

### Responde a las siguientes preguntas:

#### 1. ¿Cuál es el objeto que has tenido que crear?

Un objeto XMLHttpRequest, que es el objeto clave que permite realizar comunicaciones con el servidor en segundo plano, sin necesidad de recargar las páginas.

#### 2. Describe los valores posibles de la propiedad readyState.

Los valores definidos para la propiedad readyState son los siguientes:

| Valor | Descripción  |
|-------|--|
| 0     | No inicializado (objeto creado, pero no se ha invocado el método open).                            |
| 1     | Cargando (objeto creado, pero no se ha invocado el método send).                                   |
| 2     | Cargado (se ha invocado el método send, pero el servidor aún no ha respondido).                    |
| 3     | Interactivo (se han recibido algunos datos, aunque no se puede emplear la propiedad.responseText). |
| 4     | Completo (se han recibido todos los datos de la respuesta del servidor)                            |

### 3. Describe alguno de los valores posibles de la propiedad status (<http://tools.ietf.org/html/rfc2616#section-10>)

El primer dígito del código de estado define la clase de respuesta, las siguientes dos cifras no tienen categoría. Existen cinco valores para el primer dígito:

- 1xx: Informativo: Solicitud recibida, proceso continuo.
- 2xx: Éxito. La acción fue recibida con éxito, entendido y aceptado.
- 3xx: Redirección. Además hay que tomar medidas con el fin de completar la solicitud.
- 4xx: Error de cliente. La solicitud contiene sintaxis incorrecta o no puede cumplirse.
- 5xx: Error del servidor. El servidor no pudo cumplir con una aparentemente solicitud válida

Status-Code =

"100" ; Section 10.1.1: Continuar

"101" ; Section 10.1.2: Switching Protocols

**"200" ; Section 10.2.1: OK**

La solicitud ha tenido éxito. La información devuelta con la respuesta depende del método utilizado en la solicitud.

"201" ; Section 10.2.2: Created

"202" ; Section 10.2.3: Accepted

"203" ; Section 10.2.4: Non-Authoritative Information

"204" ; Section 10.2.5: No Content

"205" ; Section 10.2.6: Reset Content

"206" ; Section 10.2.7: Partial Content

"300" ; Section 10.3.1: Multiple Choices

"301" ; Section 10.3.2: Moved Permanently

"302" ; Section 10.3.3: Found

"303" ; Section 10.3.4: See Other

"304" ; Section 10.3.5: Not Modified

"305" ; Section 10.3.6: Use Proxy

"307" ; Section 10.3.8: Temporary Redirect

"400" ; Section 10.4.1: Bad Request  
"401" ; Section 10.4.2: Unauthorized  
"402" ; Section 10.4.3: Payment Required  
"403" ; Section 10.4.4: Forbidden  
"404" ; Section 10.4.5: Not Found  
"405" ; Section 10.4.6: Method Not Allowed  
"406" ; Section 10.4.7: Not Acceptable  
"407" ; Section 10.4.8: Proxy Authentication Required  
"408" ; Section 10.4.9: Request Time-out  
"409" ; Section 10.4.10: Conflict  
"410" ; Section 10.4.11: Gone  
"411" ; Section 10.4.12: Length Required  
"412" ; Section 10.4.13: Precondition Failed  
"413" ; Section 10.4.14: Request Entity Too Large  
"414" ; Section 10.4.15: Request-URI Too Large  
"415" ; Section 10.4.16: Unsupported Media Type  
"416" ; Section 10.4.17: Requested range not satisfiable  
"417" ; Section 10.4.18: Expectation Failed  
"500" ; Section 10.5.1: Internal Server Error  
"501" ; Section 10.5.2: Not Implemented  
"502" ; Section 10.5.3: Bad Gateway  
"503" ; Section 10.5.4: Service Unavailable  
"504" ; Section 10.5.5: Gateway Time-out  
"505" ; Section 10.5.6: HTTP Version not supported

#### 4. Describe la propiedad `responseText`

La propiedad `responseText` define el contenido de la respuesta del servidor en forma de cadena de texto.

#### 5. Describe el manejador de eventos `onreadystatechange`

El método `onreadystatechange` es responsable de manejar los eventos que se producen. Se invoca cada vez que se produce un cambio en el estado de la petición HTTP. Normalmente es una referencia a una función JavaScript.