

## 7.A. Programación AJAX en JavaScript

### 2.- Envío y recepción de datos de forma asíncrona.

#### 2.1.- Estados de una solicitud asíncrona (parte I).

Cuando se realiza una petición asíncrona, dicha petición va a pasar por diferentes estados (del 0 al 4 - propiedad `readyState` del objeto XHR), independientemente de si el fichero solicitado al servidor, se encuentre o no. Atención: dependiendo del navegador utilizado, habrá algunos estados que no son devueltos.

Cuando dicha petición termina, tendremos que comprobar cómo lo hizo, y para ello evaluamos la propiedad `status` que contiene el estado devuelto por el servidor: 200: *OK*, 404: *Not Found*, etc. Si `status` fue *OK* ya podremos comprobar, en la propiedad `responseText` o `responseXML` del objeto XHR, los datos devueltos por la petición.



[Descarga el código fuente del ejemplo dwec0721.](#) (0.01 MB)

```

////////////////////////////////////

// Función cargarAsync: carga el contenido de la url usando una petición AJAX de forma ASINCRONA.

////////////////////////////////////

function cargarAsync(url) {

    if (miXHR) {

        alert("Comenzamos la petición AJAX");

        //Si existe el objeto miXHR

        miXHR.open('GET', url, true); //Abrimos la url, true=ASINCRONA

        // En cada cambio de estado(readyState) se llamará a la función estadoPetición

        miXHR.onreadystatechange = estadoPetición;

        // Hacemos la petición al servidor. Como parámetro: null ya que los datos van por GET

        miXHR.send(null);

    }

}

////////////////////////////////////

// Función estadoPetición: será llamada en cada cambio de estado de la petición.

// Cuando el estado de la petición(readyState) ==4 quiere decir que la petición ha terminado.

// Tendremos que comprobar cómo terminó(status): == 200 encontrado, == 404 no encontrado, etc.

```

```
// A partir de ese momento podremos acceder al resultado en responseText o responseXML
```

```
////////////////////////////////////
```

```
function estadoPeticion() {

    switch(this.readyState) // Evaluamos el estado de la petición AJAX

    {        // Vamos mostrando el valor actual de readyState en cada llamada.

        case 0: document.getElementById('estado').innerHTML += "0 - Sin iniciar.<br/>";

        break;

        case 1: document.getElementById('estado').innerHTML += "1 - Cargando.<br/>";

        break;

        case 2: document.getElementById('estado').innerHTML += "2 - Cargado.<br/>";

        break;

        case 3: document.getElementById('estado').innerHTML += "3 - Interactivo.<br/>";

        break;

        case 4: document.getElementById('estado').innerHTML += "4 - Completado.";

            if (this.status == 200)        // Si el servidor ha devuelto un OK

            {

                // Escribimos la respuesta recibida de la petición AJAX en el objeto DIV

                textoDIV(document.getElementById("resultados"), this.responseText);

                alert("Terminó la petición AJAX");

            }

        break;

    } }

}
```

Créditos de la imagen

Autoría: CyborRoZ

Licencia: CC BY 2.0.

◀ DWEC07 Solucion tarea

Ir a...

7.B. Introducción a AJAX en W3Schools ▶