



7.A. Programación AJAX en JavaScript

3.- Bibliotecas cross-browser para programación AJAX.

3.4.- El método .load() y las funciones \$.post() , \$.get() y \$.getJSON() en jQuery.

La función **\$.AJAX()** es una función muy completa, y resulta bastante pesada de usar. Su uso es recomendable, para casos muy concretos, en los que tengamos que llevar un control exhaustivo de la petición AJAX. Para facilitarnos el trabajo, se crearon 3 funciones adicionales de alto nivel, que permiten realizar peticiones y gestionar las respuestas obtenidas del servidor:

El método .load()

Este método, es la forma más sencilla de obtener datos desde el servidor, ya que de forma predeterminada, los datos obtenidos son cargados en el objeto al cuál le estamos aplicando el método.

Su sintaxis es: **.load(url, [datos], [callback])**

La función **callback** es opcional, y es ahí donde pondremos la función de retorno, que será llamada una vez terminada la petición. En esa función realizaremos tareas adicionales, ya que la acción por defecto de cargar en un objeto el contenido devuelto en la petición, la realiza el propio método **load()**.

Ejemplos:

```
$("#noticias").load("feeds.HTML");

// carga en el contenedor con id noticias lo que devuelve la página feeds.HTML.

$("#objectID").load("test.php", { 'personas[]': ["Juan", "Susana"] } );

// Pasa un array de datos al servidor con el nombre de dos personas.
```

Cuando se envían datos en este método, se usará el método **POST**. Si no se envían datos en la petición, se usará el método **GET**.

Debes conocer

[Más información sobre el método .load\(\) en jQuery.](#)

La función \$.post()

Nos permite realizar peticiones AJAX al servidor, empleando el método POST. Su sintaxis es la siguiente:

\$.post(url, [datos], [callback], [tipo])

Ejemplos:

```
$.post("test.php");

$.post("test.php", { nombre: "Juana", hora: "11am" } );

$.post("test.php", function(resultados) {

    alert("Datos Cargados: " + resultados);

});
```



Debes conocer

[Más información sobre el método .post\(\) en jQuery.](#)

La función `$.get()` y `$.getJSON()`

Hacen prácticamente lo mismo que **POST**, y tienen los mismos parámetros, pero usan el método **GET** para enviar los datos al servidor. Si recibimos los datos en formato JSON, podemos emplear `$.getJSON()` en su lugar.

`$.get(url, [datos], [callback], [tipo])` `$.getJSON(url, [datos], [callback], [tipo])`

Debes conocer

[Más información sobre el método .get\(\) en jQuery.](#)

Créditos de las imágenes

Autoría: Tim Green aka atoach

Licencia: CC BY 2.0.

◀ DWEC07 Solucion tarea

Ir a...



7.B. Introducción a AJAX en W3Schools ▶