

Curso HTML5, CSS y JavaScript

Módulo 6: jQuery, Ajax y JSON

- emilianoagustingallo@gmail.com
- <https://www.linkedin.com/in/emiliano-gallo/>



¿Qué es el DOM ?

Es un **modelo de objetos** para **HTML** (Document Object Model).

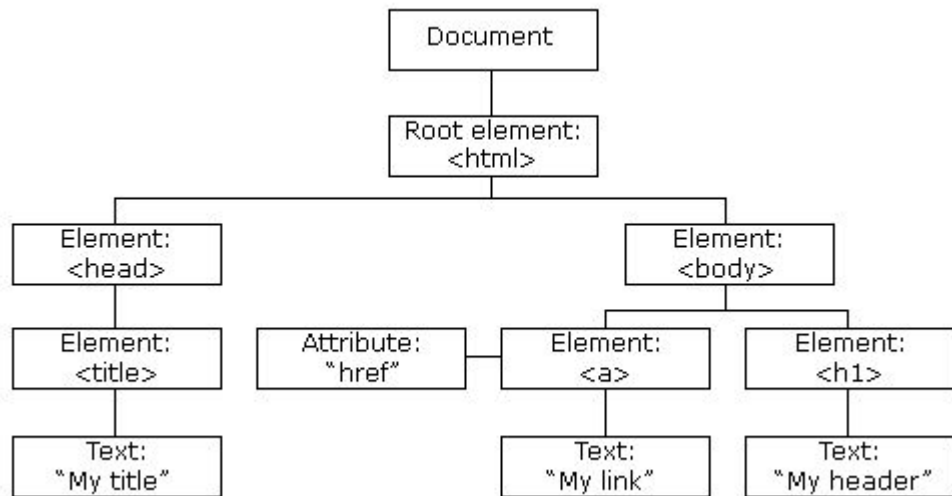
Es una plataforma e interfaz independiente del lenguaje que permite a los programas y scripts acceder y actualizar dinámicamente el contenido, la estructura y el estilo de un documento.

Define:

- Elementos HTML como objetos
- Propiedades para todos los elementos HTML
- Métodos para todos los elementos HTML
- Eventos para todos los elementos HTML

HTML DOM

Cuando se carga una página web, el navegador crea un modelo de objetos de documento (**DOM**) de la página.



Buscando elementos en DOM

Cuando se desea acceder a elementos HTML con JavaScript, primero debe buscar los elementos.

Hay algunas formas de hacer esto:

- Por **ID**: `document.getElementById("intro")`
- Por **nombre de etiqueta**: `document.getElementsByTagName("p")`
- Por **nombre de clase**: `document.getElementsByClassName("intro")`
- Mediante **selectores CSS**: `document.querySelectorAll("p.intro")`

¿Qué es jQuery?

Es una librería de JavaScript que nos ayuda a simplificar la programación de JavaScript.

jQuery toma muchas tareas comunes que requieren varias líneas de código JavaScript para llevarlas a cabo y las envuelve en métodos que se pueden llamar con una sola línea de código.

Contiene las siguientes características:

- Manipulación HTML / DOM
- Manipulación CSS
- Métodos de eventos HTML
- Efectos y animaciones
- AJAX
- Utilidades

¿Cómo usamos jQuery?

Hay 2 formas de comenzar a usar jQuery en su sitio web.

- Descargar la biblioteca de jQuery.com

```
<head>  
  <script src="jquery-3.5.1.min.js"></script>  
</head>
```

- Incluir jQuery de una CDN

```
<head>  
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
</head>
```

Primeros pasos en jQuery

La sintaxis de jQuery está hecha para seleccionar elementos HTML y realizar alguna acción.

Un ejemplo de la sintaxis: **\$(selector).action()**

- Con **\$** accedemos a jQuery
- Con **(selector)** consultamos el elemento, accedemos a él
- Con **action()** realizamos la acción

Selectores en jQuery

Nos permiten seleccionar y manipular elementos HTML. Se utilizan para "buscar" (o seleccionar) elementos HTML en función de:

- nombre
- id
- class
- tipos
- atributos
- etc

Se basa en los selectores CSS existentes y, además, tiene algunos selectores personalizados propios.

Selectores en jQuery

Nos permiten seleccionar y manipular elementos HTML. Se utilizan para "buscar" (o seleccionar) elementos HTML en función de:

- nombre: `$("elementName")`
- Id: `$("#elementId")`
- class: `$(".elementClass")`
- tipos: `$(":button")`
- atributos: `$("[href]")`

https://www.w3schools.com/jquery/jquery_selectors.asp

Algunos ejemplos

- `$(this).hide()` - oculta el elemento actual
- `$("p").hide()` - oculta todos los elementos **<p>**
- `$(".test").hide()` - oculta todos los elementos con **class="test"**
- `$("#test").hide()` - oculta todos los elementos con **id="test"**
- `$(document).ready(function(){ })` - se ejecuta al terminar de cargarse el documento

Eventos en jQuery

Son las acciones a las que puede responder una página web. Representa el momento preciso en que sucede algo.

Por ej:

- mover el mouse sobre un elemento
- seleccionando un botón de radio
- haciendo clic en un elemento

Eventos del DOM

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

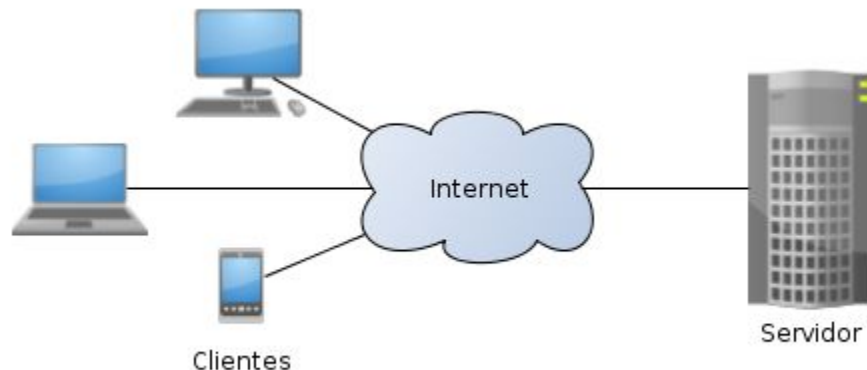
https://www.w3schools.com/jquery/jquery_ref_events.asp

Código cliente-servidor

Es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios (**servidores**) y los demandantes (**clientes**).

Un **cliente** realiza una petición a otro programa y es el **servidor** quien le da respuesta.

El **servidor** web alberga los contenidos y el **cliente** lo solicita por **HTTP**.



Código cliente-servidor

Los **contenidos estáticos**, como los elementos de HTML o imágenes, se envían al navegador y allí se visualizan.

Los **contenidos dinámicos**, como una lista de precios, un menú desplegable, funcionan **por medio de scripts**, los cuales se ejecutan e interpretan en el lado del servidor o en el lado del cliente.

Es por esto que se diferencia fundamentalmente entre los lenguajes de programación del lado servidor y los lenguajes del lado cliente.

¿Qué es AJAX?

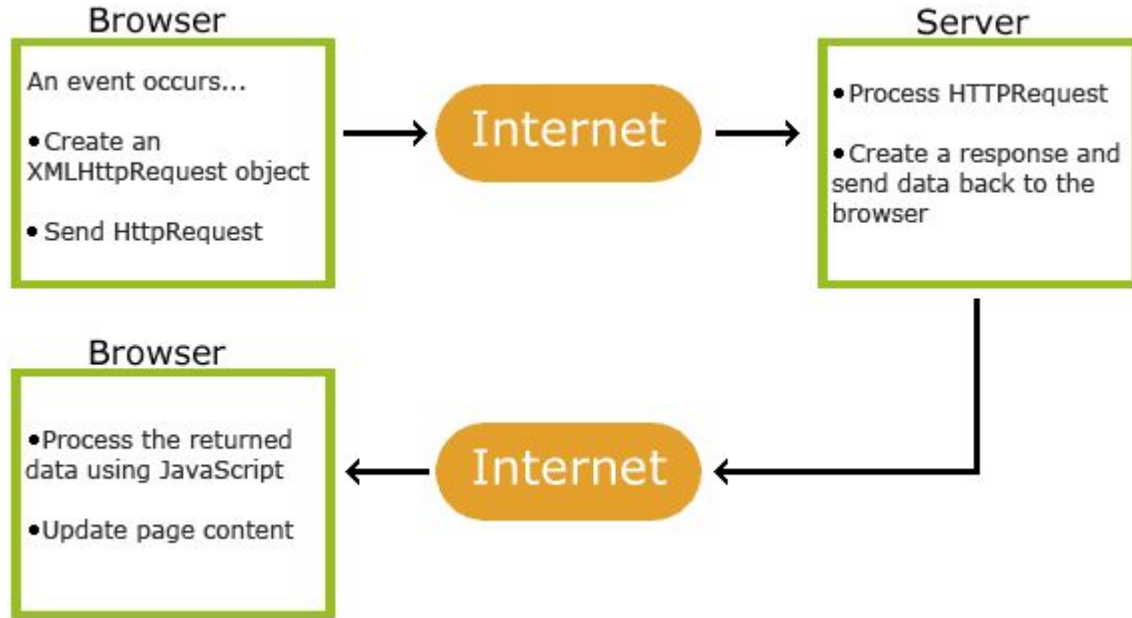
Asynchronous **J**avaScript **A**nd **X**ML

No es un lenguaje de programación.

Permite que las páginas web se actualicen de forma **asincrónica** al momento de ir al servidor a realizar una acción.

Esto significa que es posible actualizar partes de una página web, **sin recargar toda la página**.

¿Qué es AJAX?



Serializar objetos con JSON

Es un formato para **almacenar** y **transportar** datos.

Es muy útil para enviar datos desde un **servidor** a una **página web**.

- Son las siglas de JavaScript Object Notation
- Es un formato de intercambio de datos ligero
- Es independiente del idioma
- Es "auto descriptivo"

Sintaxis de un JSON

- Los datos están en pares de **nombre** / **valor**
- Los datos están separados por **comas**
- Las **llaves** sostienen objetos
- Los **corchetes** sostienen matrices

```
"employees":[  
  {"firstName":"John", "lastName":"Doe"},  
  {"firstName":"Anna", "lastName":"Smith"},  
  {"firstName":"Peter", "lastName":"Jones"}  
]
```

¿Qué es un servicio **API REST**?



Servicio API REST

API (**A**pplication **P**rogramming **I**nterface): es un conjunto de métodos / funciones que se exponen públicamente para que pueda ser utilizada por otras aplicaciones

REST: Es una **interfaz** para conectar varios sistemas basados en el protocolo **HTTP** y nos sirve para obtener y generar datos y operaciones, en formato **XML** y **JSON**.

El formato más usado en la actualidad es el formato **JSON**, ya que es más ligero y legible en comparación al formato **XML**.

Protocolo HTTP

HTTP es un protocolo que permite intercambiar información entre clientes y servidores.

Su función es permitir el intercambio de información entre **cliente** y **servidor**.

Los mensajes enviados por el cliente (por ej, desde el buscador) se llaman peticiones (**request**) y los mensajes enviados por el servidor se llaman respuestas (**response**).

Algunos ejemplos

