JavaScript & jQuery

JavaScript

JavaScript puede ser utilizado en los navegadores web para hacer a los sitios web más interactivos, interesantes y amigables.

JavaScript accede y modifica el contenido usado en una página web sin necesidad de editar el código HTML original.

Es el **único lenguaje de programación** que trabaja en el front-end.

¿Cómo funciona?

ACCESAR

Con JavaScript puede **seleccionar** un elemento, atributo o texto HTML.

MODIFICAR

Puede **añadir** elementos, atributos y texto a la página, o **remover** otros ya existentes.

PROGRAMAR

Especificar una **serie de pasos** para que el navegador web los siga (como una receta) y así accesar y modificar el contenido.

REACCIONAR

Puede indicar la ejecución de un script que debería ejecutarse tras un **evento** específico.

Ejemplos JavaScript

- Galería de imágenes
- Validación de formularios
- Recargar parte de una página
- Filtrar información

jQuery

Al contrario de JavaScript, no es un lenguaje de programación, sino que se trata de una **biblioteca** JavaScript.

Características principales:
- Selectores simples
- Tareas comunes con menos código
- Compatibilidad entre navegadores web

HTML, CSS y JavaScript

CAPA DE CONTENIDO

< html >

El HTML proporciona la estructura de la página web y añade la semántica.

Archivos **.html**.

CAPA DE PRESENTACIÓN

{ css }

CSS mejora la presentación del HTML con reglas que lo modifican.

Archivos **.css**.

CAPA DE COMPORTAMIENTO

javascript ()

JavaScript es quien cambia el cómo la página se comporta añadiendo interactividad.

Archivos **.js**.

JS Interno y Externo

ETIQUETA

<script>

La etiqueta **<script>** es usada dentro de un documento HTML para importar un archivo o definir código.

Se ejecuta donde ésta se encuentre en el HTML.

ATRIBUTO *src*

src="..."

El atributo **src** especifica la ubicación del archivo enlazado.

Sólo se requiere para JS externo.

CONTENIDO

<script></script>

El contenido de la etiqueta puede estar o no dependiendo de si se usa JS interno o externo.

La etiqueta de cierre no es opcional.

Objetos y métodos

OBJETO

document

El objeto **document** representa a la página web entera.

MIEMBROS

.write()

Se accede a **funciones** y **propiedades** con el operador **.** (punto) .

PARÁMETROS

.write("Hola")

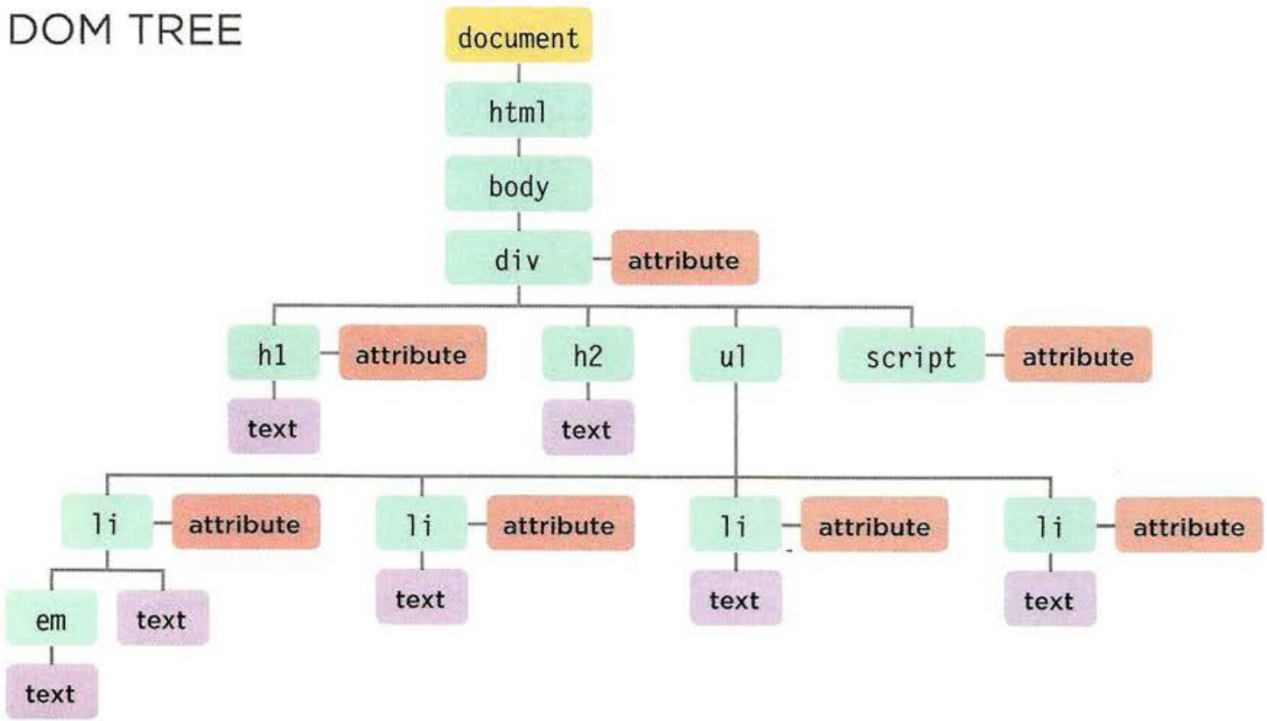
Algunos métodos requieren información para trabajar correctamente. Tal información se proporciona dentro de los paréntesis.

Document Object Model (DOM)

El DOM especifica cómo los navegadores web deberían crear un **modelo** (árbol DOM) de la página HTML, y cómo JavaScript puede acceder y modificar el contenido de ésta mientras está en la ventana del navegador.

El DOM es un objeto con métodos y propiedades (**Application Programming Interface**, **API**) que permiten realizar las funciones de JavaScript.

```
<html>
  <body>
    <div id="page">
      <h1 id="header">List</h1>
      <h2>Buy groceries</h2>
      <ul>
        <li id="one" class="hot"><em>fresh</em> figs</li>
        <li id="two" class="hot">pine nuts</li>
        <li id="three" class="hot">honey</li>
        <li id="four">balsamic vinegar</li>
      </ul>
      <script src="js/list.js"></script>
    </div>
  </body>
</html>
```



DOM Tree
By: Jon Duckett

Búsqueda de los elementos

INDIVIDUAL

$("#id")

Accede al valor de in elemento a través de su atributo **id**.

MÚLTIPLE

$(".class")

Selecciona a todos los elementos que tengan la **clase** especificada.

BÚSQUEDA

$(...).find(...)

Puede movilizarse a través del árbol DOM de un elemento (**nodo**) a otro.

- find(), closest()
- children(), siblings()

Manipulación de los elementos

TEXTO

<a>Enlace

El texto entre las etiquetas es almacenado en el **nodo texto**.

- text()
- val()

CONTENIDO HTML

...

Si el contenido entre las etiquetas es código HTML, éstos son **elementos nodo** en el árbol DOM y no texto.

- html()
- prepend(), append()
- remove(), empty()

ATRIBUTO

<a href="...">

El contenido de un atributo es almacenado en el **nodo atributo**.

- attr(), removeAttr()
- addClass(), removeClass()
- css()

Eventos

Al navegar en internet, el navegador web detecta diferentes tipos de eventos y el código puede responder a éstos.

JavaScript representa un evento mediante el objeto Event; con **target** y **type** como sus principales propiedades, y **preventDefault()** como su método más importante.

Tipos de eventos

INTERFAZ GRÁFICA

Ocurren cuando el usuario interactúa con la interfaz gráfica (User Interface, UI).

- load
- resize
- scroll

TECLADO

Ocurren cuando el usuario interactúa con el teclado.

- keydown
- keyup
- keypress

RATÓN

Ocurren cuando se interactúa con un ratón o una pantalla táctil.

- click
- dblclick
- mousedown
- mouseup
- mousemove
- mouseover
- mouseout

Tipos de eventos

FOCO

Ocurren cuando un elemento gana o pierde el foco.

- focus
- blur

FORMULARIO

Ocurren cuando el usuario interactúa con un elemento de formulario.

- change
- submit
- cut, copy, paste
- select

Escuchar eventos

SELECCIONAR NODO

$("#id")

Seleccionar el **elemento** al que se pretende asociar un escuchador de eventos.

ESPECIFICAR EVENTO

$(...).on("click")

Indicar qué **evento** se estará escuchando el nodo seleccionado.

INVOCAR CÓDIGO

$(...).on(..., f())

Indicar mediante una **función** el código que se ejecutará cuando el evento se presente.