Aplicación de JavaScript y JQuery

Aplicación de JavaScript

Su finalidad principal es permitir la creación de páginas dinámicas, con código que puede ejecutarse desde el lado cliente, alivianando la tarea del servidor y disminuyendo la cantidad de peticiones que se le realicen. Por sus características, resulta útil para validación de formularios, mostrar y aplicar efectos, y exhibir avisos en pantalla.

JavaScript es un lenguaje de scripts compacto basado en objetos (y no orientado a objetos) y sobre esos objetos podemos definir diferentes eventos.

Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la computadora del usuario.

Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del ratón sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo.

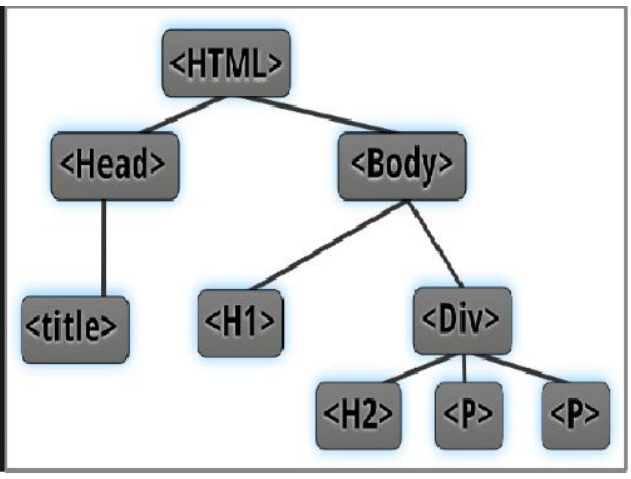
El uso de Javascript dentro de etiquetas HTML está permitido en HTML5, pero por las mismas razones que en CSS, esta clase de práctica no es recomendable. El código HTML se extiende innecesariamente y se hace difícil de mantener y actualizar. Así mismo, el código distribuido sobre todo el documento complica la construcción de aplicaciones útiles.

DOM

El **Modelo en Objetos para la Representación de Documentos** ó **DOM** (*Document Object Model*), es un estándar de objetos para representar documentos **HTML** y **XML**. A través de él, los programas pueden acceder y modificar el contenido, estructura y estilo.

En la siguiente imagen se visualiza una representación del DOM del documento HTML:

```
01. <!DOCTYPE html>
02. <html lang="es">
03. <head>
04.   <title>Título</title>
05. </head>
06. <body>
07.   <h1>Documento de ejemplo</h1>
08.   <div>
09.       <h1>Contenido dentro del div</h1>
10.       <h2>sub título</h2>
11.       <p>parrafo 1</p>
12.       <p>parrafo 2</p>
13.   </div>
14. </body>
15. </html>
```

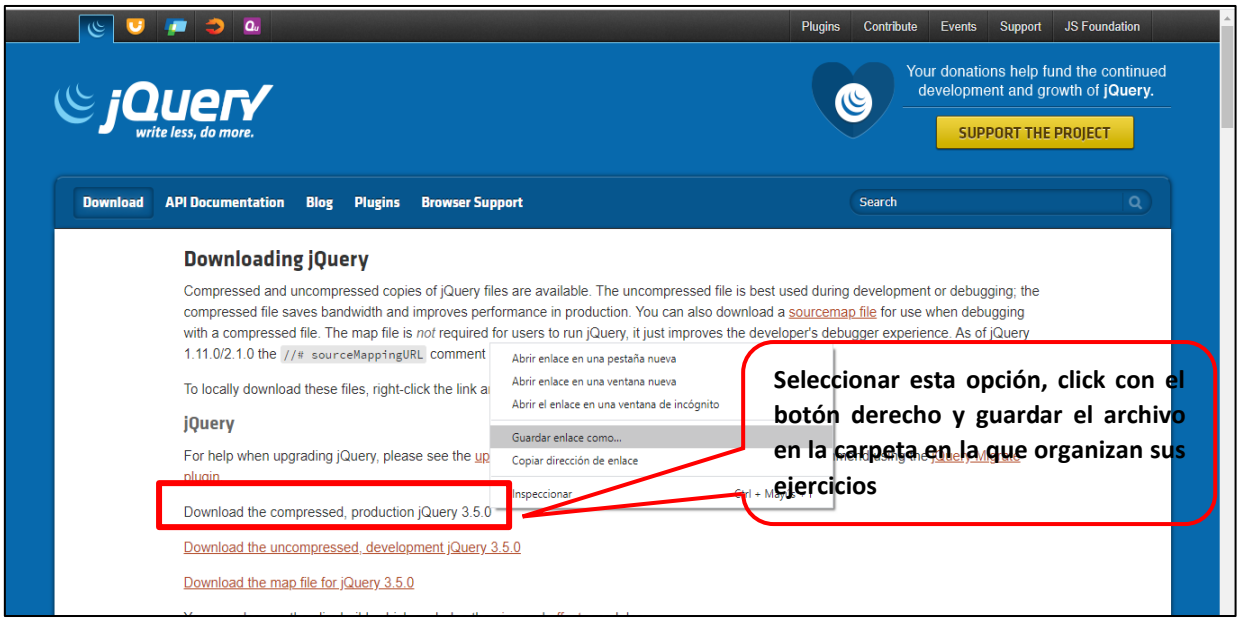


JQUERY

JQuery es una biblioteca de *JavaScript*, permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejo de eventos, desarrollar animaciones y agregar interacción con la técnica AJAX en páginas web.

¿Cómo agregar JQuery a mi documento?

En primer lugar, descargamos la librería jQuery (https://jquery.com/download/):



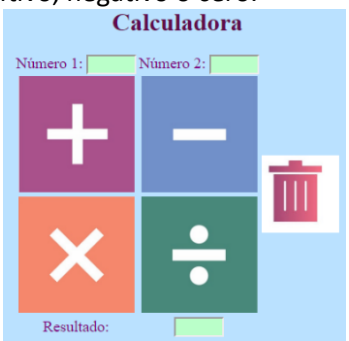
Lo siguiente es incluir la librería en nuestro documento HTML, esto lo hacemos, agregando el script dentro de la cabecera del documento (después de los archivos CSS y antes de todas las librerías que dependan de jQuery).

```
<!DOCTYPE html>
<html>
<head>
    <title> Calculadora </title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="css/estilos_caluladora.css">
    <script type="text/javascript" src="js/jquery-3.5.0.min.js"></script>
    <script type="text/javascript" src="js/calculadora.js"></script>
</head>
```

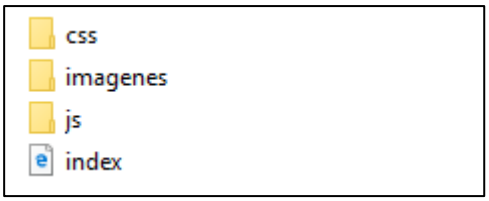
Uso de la etiqueta <script>: Para utilizar JavaScript, lo que hacemos normalmente es indicar al HTML que queremos cargar un script (una lista de órdenes) y hacerlas funcionar sobre la página actual. Para hacer esto, utilizaremos la etiqueta <script>, que permite indicar una serie de atributos:

Atributo	Valor	Descripción
src	URL	Dirección URL de un script externo a cargar.
type	tipo	Tipo de script a cargar. Si se omite, se asume **text/javascript** como valor.
charset	codificación	Codificación con la cuál se cargará el script definido en el atributo **src**.

Analizaremos el funcionamiento de JQuery sobre el siguiente ejemplo, en el que permitiremos ingresar 2 números y presionar en una imagen para realizar una operación. El resultado se mostrará en la parte inferior y además informará si es positivo, negativo o cero.



Para ello, recomiendo organizar la estructura de archivos de la siguiente manera:



En la carpeta “CSS” ubicaremos los archivos CSS, en la carpeta “js” ubicaremos todos los archivos JavaScript y en “imágenes” todas las imágenes que usemos para nuestro sitio web.

La elaboración de los archivos HTML y CSS deberán realizarla, teniendo presente que todos los estilos aplicados deber estar incluidos en el archivo CSS.
En el archivo HTML (index.html), es importante que usen el atributo **id** para identificar los objetos que posteriormente necesitaremos referenciar. Ejemplo: etiquetas correspondientes a los números, imágenes:
Requerimiento: leer el Capítulo 6: Formularios, apartados 6.1,6.2 y 6.3 del El gran libro de HTML5, para el uso de las etiquetas de formularios.

Archivo **index.html:**

```
10  <body>
11      <h1> Calculadora</h1>
12  <center>
13  <table >
14   <tr>
15      <td><label> Número 1: </label><input type="text" name="num1" id="num1"> </td>
16      <td><label> Número 2: </label><input type="text" name="num2" id="num2"> </td>
17      <td rowspan="4">  <img id="borrar" src="imagenes/borrar.jpg">
18      </td>
19   </tr>
20   <tr>
21      <td> <img id="suma" src="imagenes/suma.jpg">    </td>
22      <td> <img id="resta" src="imagenes/resta.jpg">  </td>
23   </tr>
24    <tr>
25      <td> <img id="multiplicacion" src="imagenes/multiplicacion.jpg">    </td>
26      <td> <img id="division" src="imagenes/division.jpg">    </td>
27   </tr>
28   <tr>
29      <td><label> Resultado: </label></td>
30      <td><input type="text" name="res" id="res" readonly>  </td>
31   </tr>
32  <tr>
33      <td colspan="3"><label id="txtres">  </label></td>
34   </tr>
35  </table>
36  </center>
37  </body>
38  </html>
```

Este etiquetal input es usada para mostrar el resultado y es de sólo lectura

Esta etiqueta “label” será usada para informar si el número es positivo, negativo o cero

Realizado el archivo HTML y aplicados los estilos convenientes, procedemos a la construcción del archivo “calculadora.js”

¿Cómo usar JQuery?

La selección de elementos en jQuery se hace a través de la función **jQuery([selector])**, que también podemos llamar usando el símbolo **$([selector])**. A esta función le pasaremos un selector válido, y nos devolverá un objeto desde el cual manejaremos las propiedades del DOM. jQuery admite como selectores válidos los siguientes:

- ✓ Una cadena de texto compatible con la definición de selector válido para CSS. (**$("#id")** indica un id, **$(".clase")** indica una clase)
- ✓ Elemento de DOM (**$("body")**, **$("p")**)
- ✓ Un conjunto de elementos del DOM (**$("div, span, p.clase")**)
- ✓ Un objeto jQuery; se crea un clon del objeto que tendrá los mismos objetos del DOM

Podemos decir que esta asignación viene a corresponderse con la que antes hacíamos con las funciones document.getElementById("id_elemento"), document.getElementByClass("clase").
Y de la misma forma que accedíamos a los atributos con elemento.atributo ahora usaremos las funciones definidas en jQuery para este fin. Ejemplos:

- ✓ Obtener el valor de un atributo: **$([selector]).attr("nombre_atributo")**
- ✓ Establecer el valor de un atributo: **$([selector]).attr("nombre_atributo", "valor_atributo")**
- ✓ Añadir una clase a un elemento: **$([selector]).addClass("clase")**
- ✓ Modificar estilos de CSS: **$([selector]).css("propiedad_css", "valor")**
- ✓ Obtener el contenido de una etiqueta: **$([selector]).html()**
- ✓ Manejar eventos: **$([selector]).click([función])**
- ✓ Efectos definidos en la librería de jQuery como slide, fadeIn, fadeOut, slideUp

Los principales eventos que usaremos con JQuery son:

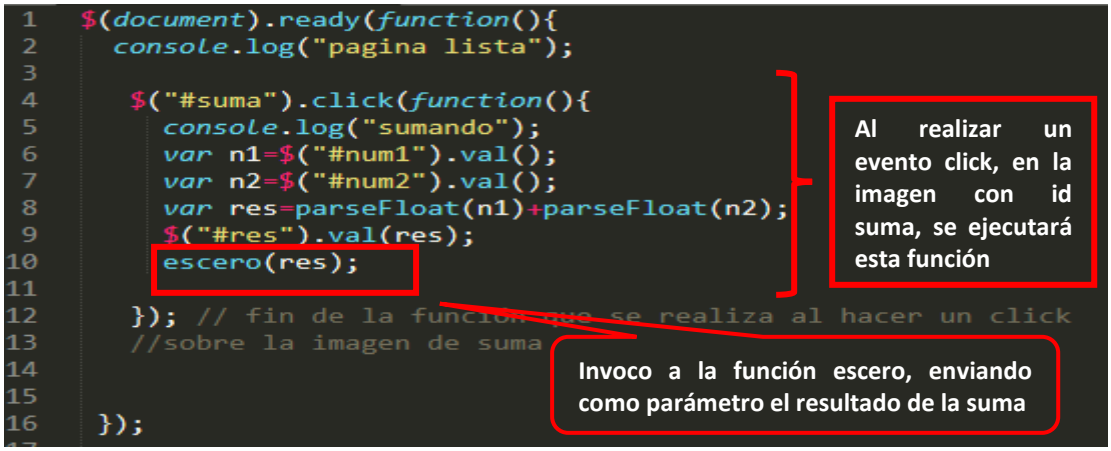
Eventos de Raton	Teclado	Formulario	Documento
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Lo primero que incluiremos en nuestro archivo JavaScript, donde utilizaremos jQuery (en nuestro caso *calculadora.js)* será la siguiente instrucción:

```
$(document).ready(function(){
  //Todo el código a ejecutar cuando el DOM está listo para recibir instrucciones.
});
```

Con **$(document)** se obtiene una referencia al documento (la página web) que se está cargando. Luego, con el método **ready()** se define un evento, que se desencadena al quedar listo el documento para realizar acciones sobre el DOM de la página.

Archivo **calculadora.js:**



val(): permite asignar u obtener valores de ciertos elementos del DOM de una web, estos elementos son los inputs de formularios y las etiquetas **select** y **textarea**.

- ✓ Sintaxis para obtener el valor de un elemento:
  `$(elemento).val()`
- ✓ Sintaxis para asignar el valor de un elemento:
  `$(elemento).val(valor)`

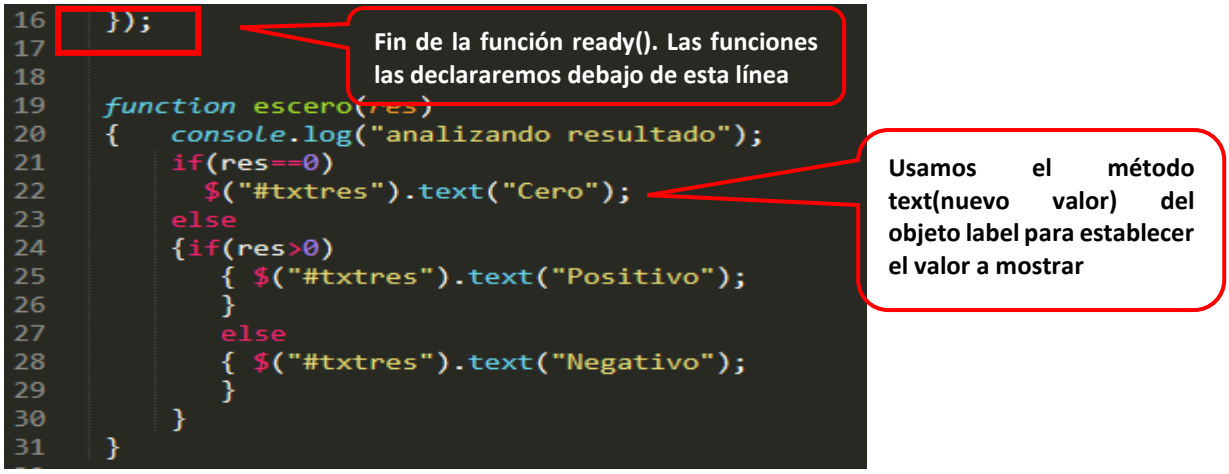
Las **funciones** y las **variables** son conceptos de JavaScript que usaremos en jQuery.

Las variables se definen con la palabra reservada "**var**", palabra reservada en JavaScript.
JavaScript es un lenguaje de tipado débil o dinámico. Esto significa que no es necesario declarar el tipo de variable antes de usarla. El tipo será determinado automáticamente cuando el programa comience a ser procesado. Esto también significa que podemos tener la misma variable con diferentes tipos.

Una función de JavaScript se define con la palabra clave "**function**", seguida de un **nombre**, seguido de paréntesis **()**. Los nombres de funciones pueden contener letras, dígitos, subrayados y signos de dólar (las mismas reglas que las variables).
Los paréntesis pueden incluir nombres de parámetros separados por comas: **(parámetro***1, parámetro2,* ... **) .** El código que se ejecutará, por la función, se coloca dentro de llaves: **{}**

```
function name(parameter1, parameter2, parameter3) {
  // code to be executed
}
```

En nuestro ejemplo de calculadora, definimos la función "**escero**", que imprimirá un texto informando si el resultado es cero, positivo o negativo:




```
16  });
17
18
19  function escero(res)
20  {   console.log("analizando resultado");
21      if(res==0)
22        $("#txtres").text("Cero");
23      else
24      {if(res>0)
25         { $("#txtres").text("Positivo");
26         }
27         else
28         { $("#txtres").text("Negativo");
29         }
30      }
31  }
```

Actividad

Programar las operaciones de suma, resta, multiplicación y división. En la división validar que el divisor sea distinto de cero. En caso de ser cero usar la función **alert()** para informar la situación.

La función **alert()** nos permite crear un mensaje de alerta sin algo más que un botón para aceptar, de modo que el mensaje desaparecerá cuando el usuario presione aceptar o cierre la ventana.

```
alert("mensaje de alerta");
```

Para la función borrar usar el siguiente código, que permite borrar los datos de los objetos input y label.

```
48  $("#borrar").click(function(){
49      console.log("borrando");
50       $("#num1").val("");
51       $("#num2").val("");
52       $("#res").val("");
53       $("#txtres").text("");
54
55    })
```