

# INTRODUCCIÓN AL DISEÑO WEB USANDO EL LENGUAJE

# HTML

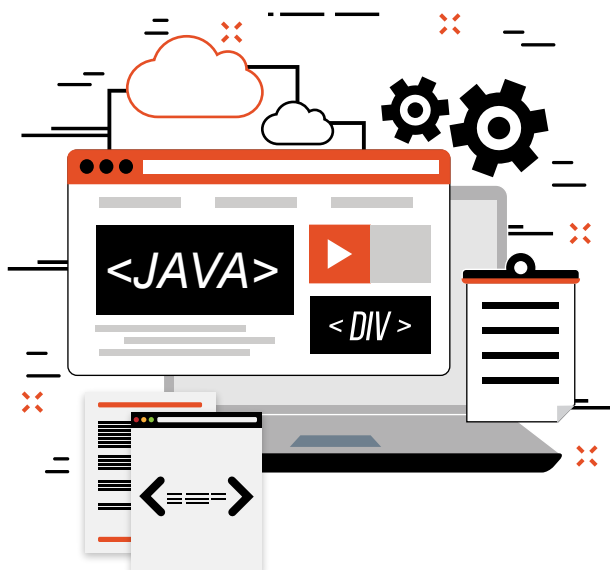


JavaScript

### 3. El lenguaje Javascript.

Javascript provee a las páginas web de interactividad, es decir, permite que se pueda comunicar con el mundo exterior. Es un lenguaje interpretado lo cual quiere decir que no requiere que sea compilado o convertido en código de máquina para ejecutarse.

Una de las principales tareas de Javascript es la validación de los datos del usuario. Por ejemplo validar que una fecha sea válida, que un correo sea válido, etc. Además puede mostrar estos errores al usuario generando una ventana con un mensaje.



Javascript es el lenguaje del navegador y como tal está incorporado dentro del mismo y no requiere otra pieza de software para funcionar. Es importante anotar que Javascript es la principal herramienta de programación del lado del cliente (Client-Side Programming) en contraposición a los lenguajes de programación del lado del servidor (Server-Side Programming) como PHP.

De la misma forma como HTML hace uso de la tecnología CSS también los archivos Javascript pueden ser incorporados a una página HTML de distintas maneras.

Una primera forma es definiendo el programa Javascript dentro de la página HTML. La segunda es definiendo un archivo separado con extensión .js e invocándolo en la página.

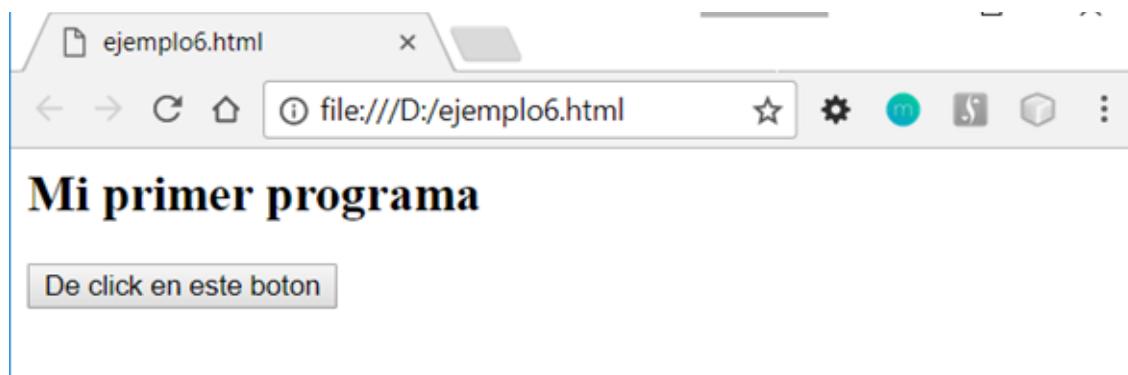
#### 3.1 Definición de un programa Javascript “inline”.

A continuación una página web que hace uso de un programa o sentencia Javascript “inline” o empotrado.

```
1 <html>
2   <body>
3     <h2>Mi primer programa</h2>
4     <button onclick="alert('hola mundo');">De clic en este botón</button>
5   </body>
6 </html>
```

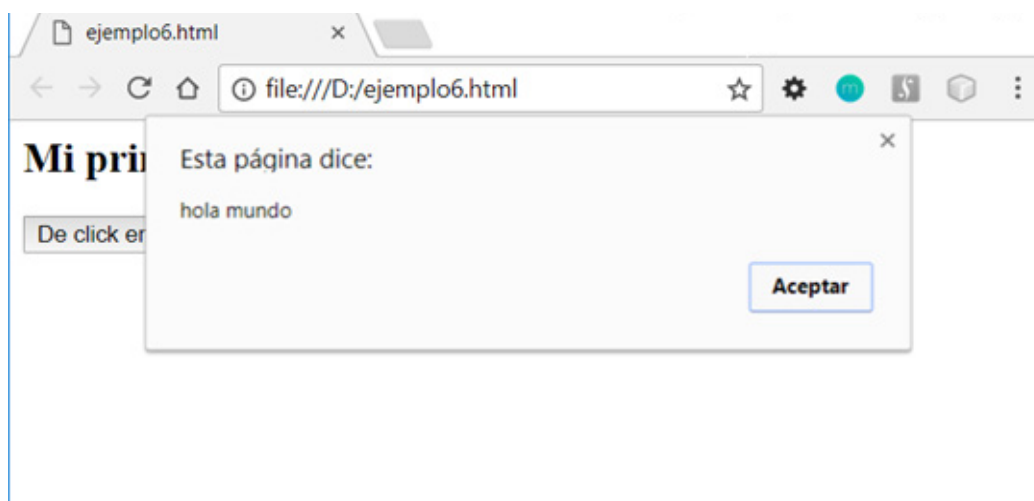
Figura 3.1. Ejemplo de Javascript “inline”.

La página generada es la siguiente:



*Figura 3.2. Ejemplo de Javascript “inline”.*

Una vez se da clic sobre el botón aparece lo siguiente:



*Figura 3.3. Página HTML con programa Javascript empotrado.*

### 3.2 Definición de un programa Javascript en una sección de la página web.

El mismo ejemplo del numeral 3.1. se puede realizar con un programa Javascript definido dentro de la misma página HTML como se muestra a continuación:

```
1 <html>
2 <body>
3   <h2>Mi primer programa</h2>
4   <button onclick="miFuncion();"> De click en este boton</button>
5 </body>
6
7 <script>
8   function miFuncion(){
9     alert('hola mundo');
10  }
11 </script>
12 </html>
```

Figura 3.4. Página HTML con programa Javascript en una sección.

En el anterior ejemplo se usó el par de etiquetas <script></script> para definir una función Javascript.

### 3.3. Definición de un programa Javascript en un archivo separado.

En este ejemplo se creó un archivo separado llamado: “programa.js” el cual se invoca en la página HTML en la sección <head> usando la expresión <script src='programa.js'></script>.

```
1 <html>
2   <head>
3     <script src='programa.js'></script>
4   </head>
5   <body>
6     <h2>Mi primer programa</h2>
7     <button onclick="miFuncion();"> De click en este boton </button>
8   </body>
9
10 </html>
```

Figura 3.5. Página HTML con programa Javascript invocado en la sección <head>

```
1 function miFuncion(){
2   alert('hola mundo');
3 }
```

Figura 3.6. Programa Javascript

### 3.4 Introducción a la programación en Javascript.

En el numeral anterior se enunciaron las distintas formas como se puede integrar un código Javascript en una página HTML.

Aplicando los patrones de diseño vistos a lo largo del curso, en especial el patrón MVC, se puede decir que la mejor forma para que las tres tecnologías interactúen, es a través de archivos de código separado. De esta forma un sitio web contiene:

- a) Archivos .html para las páginas web.
- b) Archivos .css para los estilos.
- c) Archivos .js para los programas.

La interactividad que Javascript provee a los sitios web se puede agrupar en los siguientes tipos:

- a) Implementación de un modelo de eventos.
- b) Manipulación del DOM
- c) AJAX.

#### 3.4.1. Implementación de un modelo de eventos.

Un evento es algo que ocurre durante la ejecución de un programa Javascript. Por ejemplo cuando se presiona el teclado se activa el evento “onKeyDown”, cuando se da clic en el ratón se activa el evento “onClick”, y así sucesivamente.

Si se quiere ejecutar un programa Javascript cuando se oprima clic sobre un botón se puede usar el siguiente código:

```
1
2 <button onclick="miFuncion()">Oprima este botón</button>
3
```

Los eventos también se pueden definir dinámicamente utilizando la función “addEventListener()” mediante la siguiente forma:

```
elemento.addEventListener('evento',función,boolean)
```

Donde:

Elemento: es la referencia a cualquier elemento HTML.

Evento: nombre del evento que se desea controlar.

Función: nombre de la función encargada de gestionar el evento.

Boolean: recibe el valor True o False dependiendo si se desea emplear el evento anidado con otros eventos o no. Generalmente este parámetro es False.

Ejemplo:

```
document.querySelector("#miTitulo").addEventListener('click',procesarClick,false);
```

Define que la función procesarClick se debe ejecutar al hacer clic sobre el elemento HTML identificado con el id “miTitulo”. La función “procesarClick()” debe estar definida en el archivo HTML o en un archivo .js separado.

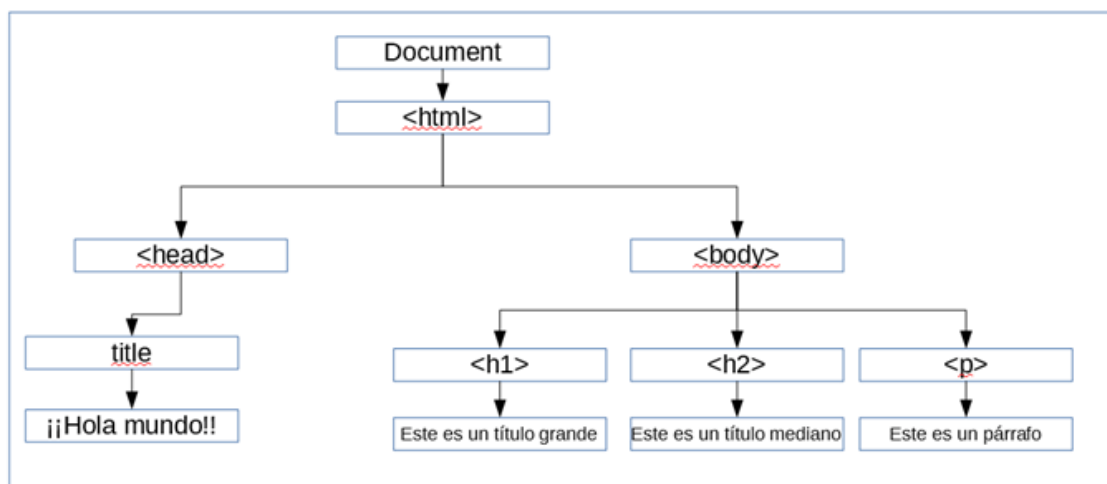
### 3.4.2. Manipulación del DOM.

El DOM es un API de programación para documentos. Una vez el navegador procesa un archivo en lenguaje HTML crea una estructura en memoria llamada el objeto Documento. Ese objeto documento tiene estructura de árbol y su elemento o nodo raíz es la etiqueta <HTML>. A partir de ahí se van anexando los demás elementos del lenguaje HTML como <head>, <body>, <h1>, <p>, <br>, <table>, <form>, etc.

A continuación un ejemplo de una página web y su correspondiente DOM.

```
1 <html>
2   <head>
3     <title>¡¡ Hola mundo !!</title>
4   </head>
5   <body>
6     <h1> Título Grande </h1>
7     <h2> Título Mediano </h2>
8     <p> Este es un párrafo </p>
9   </body>
10 </html>
```

3.7. Página HTML de ejemplo.



3.8. Ejemplo de HTML DOM.

Javascript permite acceder, agregar, modificar o borrar elementos o nodos de este árbol a través del objeto document.

Selector	Elemento	Ejemplo
getElementsByTagName	Todos los elementos de una determinada etiqueta	getElementsByTagName('p') Retoma un arreglo con todos los elementos existentes en la pagina Web. getElementsByTagName('p')[0] Accede al primer elemento tipo P de la página Web.
getElementById	Un elemento particular identificado con un determinado ID	getElementById('miID') Accede al elemento dela página Web cuyo ID es miID
getElementsByClassName	Uno o varios elementoscon un class Name determinado	getElementsByClassName('miClase') Accede a los elementos cuyo ClaseName sea miClase
querySelector	Un elemento mediante la sintaxis de seleccion de CSS	querySelector("p:first-child") Accede al primer elemento P de la página Web
querySelectorAll	Una lista de elementos que coincida con el patrón de búsqueda CSS	querySelectorAll("#principal p") Retoma un arreglo con todos los elementos P que sean hijos de principal

Para acceder a los elementos del DOM Javascript suministra las siguientes funciones:

Para crear elementos o nodos se siguen los siguientes pasos:

- Crear el nodo con el comando: `var elemento = document.createElement(nombre_del_elemento)`.
- Crear los atributos del elemento creado e integrarlos.
- Integrar el elemento creado al nodo requerido.

A continuación un ejemplo:

```

1  <!DOCTYPE html>
2  <html>
3  <div id="div1">
4      <p id="p1"este es un párrafo</p>
5      <p id="p2"este es otro párrafo</p>
6  </div>
7
8  <script>
9      var párrafo = document.createElement("p");
10     var nodo = document.createTextNode("Este es un párrafo nuevo");
11     párrafo.appendChild(nodo);
12
13     var element = document.getElementById("div1");
14     element.appendChild(párrafo);
15 </script>
16 </body>
17 </html>

```

3.9. Ejemplo de creación de elemento HTML con Javascript.

### 3.4.3. AJAX.

Los formularios web y en general las páginas web fueron concebidas para recargarse cada vez que hay un cambio en su estructura. Esto genera interrupciones que desmejoran la experiencia al usuario. Para resolver se desarrollo una tecnología basada HTML, CSS y Javascript que permite realizar consultas a servidores sin tener que recargar completamente la página.

Esta tecnología recibe el nombre de “Asynchronous Javascript and XML” o AJAX y está basada en el objeto “XMLHttpRequest”. Este recurso no aborda esta tecnología por ser un tópico avanzado.

### 3.5 Ejemplo de utilización de HTML, CSS y Javascript.

Se pueden añadir eventos a los elementos que conforman el menú del ejemplo desarrollado en este recurso. Esto se hace a través del lenguaje Javascript y en este caso se usará un archivo separado que será vinculado en la sección <head> del ejemplo.

A continuación el programa Javascript que asocia a cada opción del menú un bloque de código que se ejecuta al oprimir el botón derecho del ratón. Este programa se llamará “miPrograma.js”.

```
window.addEventListener("load",asignarEventos,false);
function asignarEventos()
{
    document.getElementById("menu1").addEventListener("click",click1,false);
    document.getElementById("menu2").addEventListener("click",click2,false);
    document.getElementById("menu3").addEventListener("click",click3,false);
}
function click1()
{
    window.alert("Hizo clic en menu 1");
}
function click2()
{
    window.alert("Hizo clic en menu 2");
}
function click3()
{
    window.alert("Hizo clic en menu 3");
}
```

3.10. Programa Javascript que asigna eventos a elementos HTML.



La siguiente es la descripción del código fuente:

Linea	Descripción
1	Al cargar la página (evento load) llama a la función asignarEventos
2.7	Accede a los elementos de la página web identificados como "menu1", "menu2" y menu"3 y le asigna al evento "clic" la ejecución de una determinada función (clic1, clic2 o clic3)
8.19	funciones que se ejecutan al hacer clic sobre alguno de los menús y que presentan un respectivo mensaje

En el código de la página web se agrega el archivo en la sección <head> así:

```

1  <header>
2      <meta charset="iso-8859">
3      <meta name="description" content="Ejemplo de Marquetación en HTML" >
4      <link rel="stylesheet" href="miEstilo2.css">
5      <script src="miPrograma.js"></script>
6  </header>

```

### 3.11. Ejemplo de inclusión de programa Javascript en la sección <head>