**Javascript - Origen**

Se popularizó con DHTML (páginas dinámicas).

La Web dejó de ser un conjunto de markup documents, para tener comportamiento de acuerdo al usuario.

DHTML = HTML + CSS + Javascript + DOM

**¿Cómo incluir un javascript?**

Conviene incluir un archivo javascript separado.

Se ejecuta su código en la línea donde se incluye.

<script type=”text/javascript” src=”js/main.js”><script>

Incluirlo al final del body, luego de que ya se cargo el html con todos sus elementos.

Se pueden agregar varios archivos JS.

**Eventos**

Un evento es algo que ocurre en el sistema, originado por el usuario u otra parte del sistema y que se avisa al sistema.

Ejemplo:

El usuario hace click.

Se terminó de cargar la página.

Pasó un segundo desde que se terminó de procesar.

Las interfases gráficas suelen programarse orientada a eventos.

**Funciones**

“Una función es un conjunto de líneas de código que realizan una tarea específica y puede retornar un valor. Las funciones pueden tomar parámetros que modifiquen su funcionamiento”.

En javascript no hay diferencia entre funciones y procedimientos.

function saludar(){

alert(“Hola”);

}

**Programación dirigida por eventos**

Un programa dirigido por eventos sigue los siguientes pasos:

Comienza la ejecución del programa.

Se llevan a cabo las inicializaciones y demás código inicial.

El programa queda bloqueado “escuchando” hasta que se produzca algún evento.

Se definen:

Eventos (click, drag, hover, load, etc).

Funciones que se ejecutan en esos eventos.

Se llama el controlador de eventos.

En la programación dirigida por eventos no sabemos la secuencia exacta de ejecución, porque se disparan diferentes códigos con diferentes acciones.

Programación secuencial: sabemos el flujo de la ejecución.

**Variables**

Una variable es un nombre que le damos a un valor que puede cambiar (o no).

El nombre no es el contenido.

Declaración de una variable

let nombre = “Pepe”;

**Constantes**

Son un nombre que le damos a un valor.

Nunca cambia con el tiempo.

Se usan para aumentar la legibilidad del programa.

Declaración de una constante

const cantDados = 2;

**Naming**

Variables se nombran con sustantivos.

Funciones comienzan con verbos.

En el caso de funciones booleanas, deben comenzar con “is”. EJ isValid()

Usar nombres descriptivos, nunca son demasiado largos.

Evitar nombres sin significado como “aux” y “temp”.

**Use Strict**

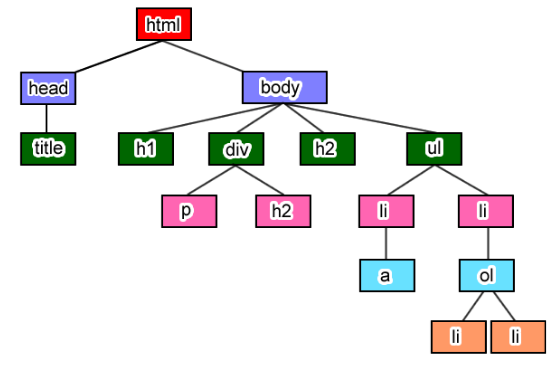
Es una buena práctica escribir al comenzar un archivo JS “use strict”;

Convierte en obligatoria la declaración de variables.

Restringe otros posibles errores de sintaxis.

**Arbol HTML**

Una manera de comprender las dependencias y relaciones entre elementos es mediante un diagrama de árbol.



**DOM**

El document Object Model es una API (Application Programming Interface) para documentos HTML y XML.

Representación estructurada del documento.

Permite modificar el contenido

Es lo que conecta las páginas web con Javascript.

El DOM es un árbol de objetos.

Obtener nodos del DOM mediante ID:

let elem = document.getElementById(“identificador”);

Al DOM podemos:

Agregarle nodos (Etiquetas).

Editar nodos (Es como cambiar el HTML).

Borrar nodos (Es como borrar etiquetas).

**Generar números aleatorios**

Se utiliza Math.random();

Podemos utilizar también la función matemática Math.floor(); que redondea al menor entero.

let numeroAleatorio = Math.floor(Math.random() \* 5);

**Identificamos elementos por etiquetas**

document.**getElementsByTagName**(“TAG”).[¿Qué Tag?].onclick = nombreFunción;

**Identificamos un único elemento**

document.**getElementById**(“id”).onclick = nombreFunción;

**Identificar elementos por clase**

document.**getElementsByClassName**(“”).onclick = nombreFuncion;

**Nuevos métodos para JavaScript**

**querySelector()**

Retorna el primer elemento que corresponde al grupo de selectores indicado entre paréntesis. Dentro del paréntesis se especifican selectores CSS como por ejemplo “.important”, “#principal”.

**querySelectorAll()**

Retorna todo los elementos que corresponden al grupo de selectores indicado entre paréntesis. El valor devuelto es un array conteniendo los elementos en el mismo orden en el que aparecen en el documento.

**Eventos**

**addEventListener()**

Estándar de la especificación HTML5

El método tiene 3 argumentos:

Nombre del evento

Función a ejecutar

Valor booleano

EJ:

addEventListener(“evento”, función, valor booleano)

Eventos comunes:

click: Click sobre algo.

mouseover: Mouse encima de algo.

mouseout: Mouse fuera de algo.