w3school.com

https://www.w3schools.com/jsref/dom\_obj\_event.asp

JavaScript esta categorizado como un lenguaje enfocado al desarrollo web y es el encargado de transformar nuestros proyectos webs 'estaticos' en sitios webs 'interactivos'.

IDE=entornos de desarrollo integrado.

JavaScript es uno de los pilares. La parte logica

html la parte de los contenidos

CSS la parte de los estilos

aplicado a una casa el html es la estructura, las paredes, la separacion de habitaciones y lo demas. el CSS son los colores y como estara definida y el JS es la parte logico como el on/off de las luces

HTML/CSS lenguajes de marcado. Crear y definir elementos.

JavaScript lenguaje de programacion. Definir la logica , el comportamiento.

diferencias entre Java y JavaScript: Ambos lenguajes no tienen relacion entre si.

sus objetivos, entorno y funciones son distintos.

lo unico que tienen en comun, son su 4 letras.

Permite procesar informacion Offline/Online

Cargas asincronicas (por partes) de documentos (AJAX)

Modificar cualquier parte de un documento

Acceso total a HTML5

Funciona en el lado del cliente

Solo es necesario un browser (navegador de internet) y un editor de texto.

INDEX es la parte principal, siempre el primer documento sera index.

Expresion: un fragmento de codigo que produce un valor.

Ejemplo: 24, "Hola Mundo", 1+2, typeof"bueno"

Declaracion: una sentencia completa.

Ejemplo: console.log(1\*2);

"</br>" salto de linea

PROGRAMACION ORIENTADA A OBJETOS

Propiedades(variables): Es una caracteristica especial de un objeto.

Metodos(funciones): es la accion o funcion que tiene un objeto.

Clases: es una abstraccion de lo que es.

Instanciar una clase: cuando vamos a crear un objeto.

Herencia: heredar propiedades principales, pero diferenciables.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Manera de proceder

-crear una clase

-instaciar o crear un objeto

-meter datos

-llamar datos

Ejemplo:

CLASE mascota{

PROPIEDADES (variables)

cabeza

patas

cola

METODOS (funciones)

camina

corre

come

}

OBJETO gato uno{

cabeza: grande

patas: largas

cola: pequeña

camina: rapido

corre: largas distancias

come: comida especial

}

OBJETO gato dos{

cabeza: pequeña

patas: cortas

cola: larga

camina: lento

corre: no

come: cualquier comida

}

mostrame gato uno > cabeza

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DOM = Document Object Model. Como definir la estructura dentro de nuestro sitio web, aprender a modificar los objetos dentro del documento HTML con JS, como resultado de esto se crea un a web interactiva.

en consola puedes acceder con el nombre de la variable a las etiquetas

con la variable como array a la posicion de las etiquetas

.innerHTML : acceder a todo el contenido de las etiquetas

.innerHTML ="cuinl" //reemplazar el contenido de las etiquetas

Ejemplo:

elementosP

elementosP[0]

elementosP[0].innerHTML

elementosP[0].innerHTML="Hola"

1.- Introduccion al DOM Document object model.

2.- Cada uno de estos elementos son nodos que podemos acceder en el DOM Document object model.

3.- Para acceder al DOM Document object model seleccionando todos lo elementos mediante etiquetas. Otra forma de seleccionar los elementos, es seleccionar mediante los ID.

<br> salto de linea

<b></b> negritas

<i></i> cursiva

width: anchura

height: altura

background: fondo

display: inline-block; pantalla: bloque en linea

margin-right: margen derecho

border: borde

padding: relleno

font-size: Tamaño de fuente

AJAX: Asynchronous JavaScript And XML (JavaScript asincrono y XML)

Cliente Navegador

Servidor Hosting

XMLHttpRequest: Es utilizado para intercambiar datos detras de escenas.

Grac ias a este objeto es posible actualizar distintas partes de una pagina web sin recargar.

Como se crea: var xhr = new XMLHttpRequest;

Metodos disponibles en XMLHttpRequest

new ->Crea un nuevo request

.abort() ->Cancela el request

.getAllResponseHeaders() ->Retorna informacion del header

.open() ->Informacion basica del request

.send() ->Envia el request al servidor

.setRequestHeader() ->Agrega valores al header que se desea enviar

Ejemplo:

var xhr = new XMLHttpRequest;

xhr.open("POST","url",true);

xhr.send();

POST o GET

POST: Utilizado para enviar o intercambiar informacion.

GET: Utilizado para obtener informacion del servidor.

Estados y propiedades de las solicitudes

Propiedades disponibles en XMLHttpRequest

readyState=El estado del XMLHttpRequest

0=request sin iniciar

1=conexion establecida

2=request recibido

3=procesando el request

4=request finalizado y la respuesta esta lista

onreadystatechange=define una funcion que debera ser llamada cuando el estado cambie.

esta funcion se ejecuta cada que readyState cambia.

responseText=retorna la respuesta como string

responseXML=retorna la respuesta como XML

status=retorna el estado del request:

200: "Correcto/ok"

403: "Prohibido/Forbidden"

404: "No encontrado/Not Found"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cookies: un pequeño archivo de texto que se almacena en el ordenador, cuando se navega en una pagina web.

Almacenamiento Local con HTML%

Almacenamiento Local vs Cookies

- El almacenamiento local nos permite almacenar, normalmente unos 5MB, las cookies solo 4KB.

- En la comunicacion cliente/servidor, las cookies almacenadas en el cliente son transmitidas en la cabecera de cada peticion al servidor. Esto no sucede con LocalStorage.