Temas para el

Desarrollo de páginas web

Índice

HTML

CSS  
JAVASCRIPT  
CSS GRID

HTML

<H1 al H6></H?> crea texto que se utiliza para título.

Solo se puede utilizar un h1 para que el navegador no te penalice.

<p></p> se utiliza para añadir párrafos.

<br/> Salto de línea.

<b></b> Texto dentro de él lo pone en negritas.

<strong></strong> Lo pone en negritas y el texto dentro de él sirve para que los buscadores de énfasis a ese texto.

<span></span> sirve para que los buscadores de énfasis a ese texto.

<i></i> pone el texto en cursiva.

<blockquote></blockquote> Una cita de un libro o página.

<cite></cite> Nombre del autor.

<hr/> Agrega una línea que divide los elementos.

<a href=”link”></a> crea un texto que si le damos clic nos envía a otra pagina.

<a target=”\_blank”></a> Envía al link pero en una nueva pestaña

<link rel=”stylesheet” href=”hoja.css” type=”text/css”> agrega estilo css a la página web.

<script src=”link del archivo JavaScript></script> agrega código para ejecutar de js.

<link rel=”shortcut icon” type=”image/x-icon” agrega un icono en la pestaña de tu pagina

href=”nombre.ico”> Sirve para agregar un icono en la pestaña de la página.

<ul></ul> lista desordenada.

<li></li> Cada elemento de la lista.

<ol></ol> lista ordenada muestra el número de cada elemento.

<img src=”ruta de imagen”/> agrega una imagen .

Width=”” ancho de la imagen.

Height=”” altura de la imagen.

Alt=”” agrega un texto que identifica a la imagen para que mejore tu página y así el navegador lo muestre en listas mejores.

Title=”” agrega el texto cuando pasas la flecha del mouse.

<table></table> Crea una tabla.

<tr></tr> crea las filas.

<td></td> crea las columnas.

<form></form> crea un formulario.

Action=”” base de datos que tendrá el formulario.

Method=”” método por donde se obtendrán los datos ya sea GET o POST.

<input> tipo de entrada para el formulario

Type=”” tipo de entrada que será.

* Text: Agregar texto
* Password: Para agregar contraseñas.
* Email: Comprobar si un correo es correcto.
* Checkbox: Agrega un cuadro para marcar una casilla.
* Radio: Agrega un cuadro para marcar una casilla.
* Button: Agrega un botón.
* Submit: Envía los datos obtenidos.
* Color: Escoge un color
* Range: Agrega una barra.

Value=”” Agrega un texto predeterminado al input.

Placeholder=”” Agrega un texto predeterminado al input pero borra el texto dicho.

Maxlength=”” numero de caracteres máximos que puede ingresar el usuario.

Name=””identifica al input para la base de datos

Id=”” agrega una palabra clave a la etiqueta para poder modificarla

Class=”” agrega una palabra clave a la etiqueta para poder modificarla

<label for=”id del input”></label> texto relacionado con el input.

<select></select> Agrega una lista despegable.

<option></option> Agrega un elemento a la lista despegable.

<header></header>Encabezado.

<section></section> Crea secciones y divide la pagina

<article></article>puede agregar dentro un articulo o información importante de tu pagina.

<aside></aside> sirve para contener información no muy relevante.

<div></div> divide la información en “carpetas”.

<footer></footer> Pie de página.

<meta/> Sirve para el posicionamiento en buscadores.

Name=””description

Content=””El contenido que tiene tu página puedes poner información acerca de tu página. Máximo 150 caracteres.

CSS

***Sintaxis:***

A{

Todos los elementos “a” serán modificados por los parámetros que pongamos aquí.

Color: White;

Font-size: 30px;

Background:aquamarine;

}

P a{

Los elementos de a(links) que estén dentro de p(Parrafo) serán modificados.

}

A[href=<http://www.google.com.mx>]{

Los elementos de a(links) que tengan esta dirección serán modificados.

}

***Formatos:***

Font-size: Valor px, em, rem, %(medida relativas).

Px= pixeles

Em= cantidad de pixeles por default \* cantidad de em;

Rem= toma los pixeles del primer Font size \* cantidad de rem.

Color cambia el color de la letra.

Background cambia el fondo.

Url(link de la imagen para ponerla de fondo);

Background-repeat: si deseas repetir la imagen de fondo.

Background-attachment: (para fijar la imagen como fondo y no se mueva).

Width cambia el ancho del parámetro (px, %, em, rem)

height cambia el alto del parámetro (px, %, em, rem)

margin cambia el marge externo del parámetro (px(arriba, abajo), px(de los lados))

padding cambia el marge interno del parámetro (px(arriba, abajo), px(de los lados))

border agrega un contorno al parámetro( solid #color).

Display modifica el tipo de parámetro ya sea en línea o en bloque.

Position :relative (Para modificar la posición de un parámetro).

Top : #px desplazara el objeto de arriba hacia abajo.

Left : #px desplazara el objeto de izquierda a derecha.

Right : #px desplazara el objeto de derecha a izquierda.

***Formato de letras:***

Font-family: Cambia la tipografía del texto.

Font.size: Tamaño del texto.

Text-weight: bold(pone la las letras en negritas).

Text-aling: center( alinea el texto ya sea centrado).

Line-height: Altura del renglón del texto.

Text-decoration: line-trough (Aplica una línea partiendo el texto).

Text-Transform: capitalize(Cada inicial de cada letra la pone en mayúscula) uppercase(Todo en mayúscula) lowercase(Todo en minúsculas).

Letter-spacing: espaciado entre letras.

Word-spacing: espacio entre palabras.

Text-shadow: agrega sombra a las letras (#px #px #px #color).

***Formato de links:***

Font-size: tamaño del link.

Text-decoration : none quita lo subrayado

Color: color del link.

Margin: #px(arriba) #px(derecha) #px(abajo) #px(izquierda).

***Estados de links:***

a:link{

color: #color(le da un color especial a los sitios que no hemos visitado aun).

}

a:hover{

Atributos que le quieras dar cuando el mouse pase por el parámetro.

}

a:active{

Atributos que le quieres dar cuando le damos clic al parámetro.

}

a:visited{

Atributos que le quieres dar a enlaces que ya hemos visitado.

}

***Links(Apariencia de botón).***

a{

background: aquamarine;

color: #fff; (blanco).

Padding: 10px 20px;

Display:inline-block;

Box-shadow: 0px 5px 0px #000;

}

a:hover{

Box-shadow: 0px 2 px 0px #000;

}

a:active{

box-shadow:none;

}

***Imagenes.***

Img{

Height:0px; altura de la imagen.

Width: 0px; ancho de la imagen.

Display: block;

Border-radius: #px (Redondea la imagen).

Box- shadow: 5px 5px 0px #ccc; (agrega una sombra a la imagen).

Padding: 10 px;

Border: 1px solid #fff (Agrega como un marco a la imagen).

}

.thumb{

Transition: 300ms all esae; (Animacion tipo zoom)

}

***Listas.***

Ul{

List-style-type: disc(círculos rellenos), circle(circulo sin relleno), square( cuadrados).

List-style-position: inside(agrega un tabulador),outside(valor por defecto).

}

Ol{

List-style-type: decimal ,decimal-leading-zero(agrega un 0 para complementar el tamaño), upper-latin(letras mayúsculas), lower-latin(letras minúsculas)

}

***Menu.***

.menu{

Background: #ccc;

Width: 100%;

List-style:none;

Margin:0;

Padding:0;

}

.menu li{

Dispay:inline-block;

}

.menu li a{

Padding: 10px 20px;

Display: inline-block;

Color: #fff;

}

.menu li a:hover{

Background: #ccc;

Color:#000;

}

***Tablas.***

Table{

Border: 1px solid #000;

Border-collapse: separate(darle separación), collapse(los bordes se juntan).

Witdh: #px;

Height: #px;

}

Table td{

Border: 1px solid #000;

Padding: 20px;

}

Tablet tr:hover{

Background:#ccc;

}

Table td:hover{

Background:#000;

Color: #fff;

}

***Formularios.***

Form{

Width: 25%;

Border: 1px solid #ccc;

Margin: 20px;

Padding: 10px;

}

Label{

Font-size: 14px;

Display:block;

Width:100%;

}

Input, textarea{

Margin-bottom:20px;

Width:100%;

Padding:10px;

Box-sizing: border-box;

Border:1px solid #ccc;

}

Input:focus, textarea:focus{

Border:1px solid #1668C4;

}

Textarea{

Resize:vertical(podrá crecer hacia abajo).

Max-height: 300px;

Min-height: 100px;

}

Input[type=”submit”]{

Margin-bottom:0px;

Background:#1668C4;

Color:#fff;

Border:none;

}

Input[type=”submit”]:hover{

Background: #4ª88F4;

Cursor:pointer;

}

body::-webkit-scrollbar {

/\* Sirve para poder modificar el fondo a la barra de navegación\*/

    width: 10px;

    background: #021b79;

}

body::-webkit-scrollbar-thumb {

/\* Sirve para poder modificar la barra de navegacion \*/

    background: rgba(255,255,255, .2);

    border-radius: 10px;

    border-right: 2px solid #021b79;

}

JavaScript

***Mandar a llamar.***

<script src=”Link”></script>

***Variables***

Var nombre de la variable;

Nombre de la variable (no es necesario agregar var todo el tiempo solo basta con una y agregar un tabulador y una “,”);

***Concatenar información.***

(variable + “Texto“ + variable);

***Palabras clave.***

Alert(“Texto que aparecerá en pantalla(o variable)”);

Document.write(“Texto que aparecerá en pantalla(o variable)”);

Console.log(“Texto que aparecerá en la consola(o variable)”);

Prompt(“Texto que aparecerá en pantalla(o variable)”,”Donde se guardara temporalmente.”);

Variable =Prompt(“Texto que aparecerá en pantalla(o variable)”);

***Tipos de datos.***

Numericos – Flotantes o enteros.

Var edad = 19;

String - cadena de texto.

Var nombre = “Pablo”;

Var apellido = “Hernandez”;

Document.write(nombre + “ “ + apellido); concatena las variables.

Var texto= “los mejores videos son de \”FalconMaster\” “.

Array – Arreglo.

Var amigos = [“Pablo”, “Cesar”, “Edgar”

];

Document.write(amigos[posicion]);

Object – Objeto.

Var objeto = {

Color: “Black”,

Size: 12

}

Boolean – Boleano.

True

False

Undefined - Indefinido

Var a;

Null - Nulo

Var n = 10;

Var n = null; borrara el valor de la variable.

NaN

Var t = “Hola”;

Documet.write(t \* 2); no se puede multiplicar una cadena de texto.

***Arreglos.***

Var nombre = [“Carlos”, “Juan”, “Juanita”];

Método de acceso.

Document.write (nombre [posición]);

Cambiar el valor de una posición.

Nombre [0] = “Arturo”;

Cantidad de elementos de un arreglo.

Document.write (Nombre.length );

Agregar elementos a un arreglo.

Nombre [nombre.length] = “Pepe”;

Nombre.push(“jose”,”oscar”,”Fernando”);

Eliminar elementos de un arreglo.

Nombre.pop();

Concatenar arreglos.

Var nombre = [“Carlos”, “Juan”, “Juanita”];

Var nombre2 = [“Pepe”, ”Enrique”, ”Gustavo”];

Var nombre3 = nombre.concat(nombre2);

Documen.write(nombre3);

Dividir arreglos;

Document.write (amigos.join(“ : “);

***Condicionales.***

If (condición) {

Instrucciones a ejecutar si se cumple la condicion.

}

Else{

Instrucciones a ejecutar si no se cumple la condición.

}

***Ciclo For.***

For(var i=0;i<valor; i++){

Instrucciones a ejecutar si se cumple la condicion.

}

***Ciclo While.***

While(i<=valor){

Instrucciones a ejecutar si se cumple la condicion.

i++;

}

***Ciclo Do While.***

Do{

Instrucciones a ejecutar si se cumple la condicion.

i++;

}while(i<=valor);

***Funciones.***

Fuction nombre(parámetros){

Instrucciones a ejecutar

Return valor;

}

Mandar a llamar la función

Nombre(parámetro);

***Funciones y formularios.***

Convertir un tipo de dato a otro.

ParseInt

ParseFloat

Obtener información a través de un formulario

<input type=”text” id=”valor”>

<script>

Var valor = document.getElementById(“valor”).value;

</script>

***Métodos y Propiedades para cadenas de texto.***

Var texto = “Pepe”;

Texto.length;

Cuantos caracteres tiene la cadena de texto.

Texto.substring(cantidad).

Elimina caracteres de izquierda a derecha.

Texto.substr(posición inicial, posición final).

Agarra los caracteres de la posición inicial hasta la final.

Texto.indexOf(“letra”, posición donde queremos que inicia a verficiar);

Posición de la primera letra que tiene en la cadena de izquierda a derecha.

Texto.lastindexOf(“letra”,Posicion donde queremos que inicia verificar=;

Posicion de la primera letra que tiene la letra de derecha a izquierda.

Texto.replace(“Palabra a remplazar”, “nueva palabra”);

Texto.toLowerCase();

Convierte a minúsculas toda la cadena.

***DOM(Document Object Model).***

Var elementosP = document.getElementsByTagName(“p”);

La variable almacenara en un arreglo cada etiqueta P de nuestro sitio web.

Var variable = document.getElementsById(“id de la etiqueta”);

La variable almacenara la etiqueta con la id obtenida de nuestro sitio web.

***Crear Nodos del DOM.***

<body>

<div id=”subtitulo”></div>

</body>

En <script>:

Crear el elemento

Var elemento = document.createElement(“h2”);

Crear un nodo de texto.

Var contenido = document.createTextNode(“Contenido”);

Añadir el nodo de texto al elemento

Elemento.appendChild(contenido);

Agregar atributos al elemento

Elemento.setAttribute(“align”,”center”);

Agregar el elemento al documento;

Document.getElementById(“subtitulo”).appendChild(elemento);

***Agregar Nodos del DOM.***

Var elemento = document.createElement(“li”);

Var contenido = document.createTextNode(“Contenido”);

Elemento.appendChild(contenido);

Agregamos una variable padre

Var padre= document.getElementByTagName(“li”)[0].parentNode,

Hijo= document.getElementByTagName(“li”)[0];

padre.appendChild(elemento); Agrega el elemento al final de todos los elementos iguales de TODO el documento.

padre.insertBefore(elemento, hijo);

Agrega el elemento antes de la posición del elemento dado.

***Modificar, remplaza y eliminar Nodos del DOM.***

<li id=” uno” >Lorem ipsum dolor <li>

En <script>:

Var hijo = document.getElementById(“primero”);

Modifica el texto de la etiqueta seleccionada y puedes agregar etiquetas dentro del nuevo contenido.

PrimerElemento.innerHTML =” Nuevo contenido”;

Modifica el texto de la etiqueta seleccionada.

PrimerElemento.textContent=”Nuevo contenido”;

Var elemento = document.createElement(“li”);

Var contenido = document.createTextNode(“Contenido”);

Elemento.appendChild(contenido);

Var padre= document.getElementByTagName(“li”)[0].parentNode;

Referencia = document.getElementByTagName(“li”)[0];

Remplaza la información del elemento hecho con el nuevo elemento.

Padre.replaceChild(elemento,referencia);

Elimina el elemento dentro de los paréntesis

Padre.removeChild(referencia);

***Atributos del DOM.***

.azul{

Background:blue;

}

Var elemento= document.createElement(“div”),

Padre= document.getElementById(“contenedor”),

Referencia= document.GetElementsByTagName(“div”)[0];

padre.insertBefore(elemento,referencia);

Elemento.setAttribute(“class”,”azul”);

***Validación de formulario.***

var formulario = document.getElementsByName('formulario')[0],

             elementos = formulario.elements,

             boton = document.getElementById("btn");

Creo una variable que tome todos los elementos del formulario, otra que los organiza y el botón.

        var validarNombre = function(e){

            if(formulario.nombre.value == 0){

                alert("Completa el campo nombre");

Si el campo nombre esta vacio envía una alerta.

                e.preventDefault();

No envía ninguna información ya que el campo esta vacion

            }

        };

        var validarOpcion =function(e){

if(formulario.sexo[0].checked == true || formulario.sexo[1].checked == true){

Si algún botón de sexo esta activo no ocurre nada.

            }else{

                alert("Completa el campo sexo");

Si no enviara un mensaje.

                e.preventDefault();

            }

        };

        var validarTerminos = function(e){

            if(formulario.terminos.checked == false){

Si no acepta los términos envía un mensaje.

                alert("Acepta los terminos y condiciones");

                e.preventDefault();

            }

        };

        var validar = function(e){

            validarNombre(e);

            validarOpcion(e);

            validarTerminos(e);

Realiza cada function en paralelo. Se agrega la e para poder usar el preventDefault();

        };

        formulario.addEventListener("submit", validar);

Crea un evento cuando enviamos los datos y agregamos el evento que queremos que realice.

***Aplicación de creación de tareas con Js***

(function(){

    //variables

    var lista = document.getElementById("lista"),

        tareaInput = document.getElementById("tareaInput"),

        btnNuevaTarea = document.getElementById("btn-agregar");

    //funciones

    var agregarTarea = function(){

        var tarea = tareaInput.value,

            //variable que guarda lo que el usuario ingrese.

            nuevaTarea = document.createElement("li"),

            //crea una variable para crear un elemento <li>

            enlace = document.createElement("a"),

            //crea una variable para crear un elemento <aZ

            contenido = document.createTextNode(tarea);

            //crea una variable que guarda como texto la variable tarea

        if(tarea == ""){

            tareaInput.setAttribute("placeholder","Agrega una tarea valida");

            tareaInput.className = "error";

            return false;

            //Si no ingresa nada el usuario envia el mensaje de agregar una tarea valida y le atribuye la clase error y termina la funcion.

        }

        enlace.appendChild(contenido);

        //agrega el contenido al enlace

        enlace.setAttribute("href","#");

        //agrega el atributo href para que envie a una direccion que es asi misma.

        nuevaTarea.appendChild(enlace);

        //Agrega el enlace dentro de un <li>

        lista.appendChild(nuevaTarea);

        //Agregamos la nuevaTarea a lista padre de todas nuestros <li> al final

        tareaInput.value= "";

        //Al momento que enviamos la tarea a la lista el input se vacia para que el usuario pueda agregar una nueva tarea y no tenga que borrar la anterior.

        for(var i = 0; i < lista.children.length; i++){

            lista.children[i].addEventListener("click",function(){

                this.parentNode.removeChild(this);

                //cuando un elemento es seleccionado se removera el mismo.

            });

        }

    };

    var comprobarInput = function(){

        tareaInput.className = "";

        tareaInput.setAttribute("placeholder","Agrega tu tarea");

        //Al momento de ingresar al input agregar el texto o ya sea si estaba vacio y mando el error lo limpia

    };

    var eliminarTarea = function(){

        this.parentNode.removeChild(this);

        //cuando el elemento sea seleccionado este mismo se removera.

    };

    //Eventos

    btnNuevaTarea.addEventListener("click",agregarTarea);

    tareaInput.addEventListener("click",comprobarInput);

    for(var i = 0; i < lista.children.length; i++){

        lista.children[i].addEventListener("click",eliminarTarea);

//Comprueba la cantidad de elementos que hay en la lista

    }

}());

***Timers, Timeout e intervalo.***

   //Variables

            var contador = 0;

            var saludo = function(){

                contador++;

                console.log(contador);

                if(contador === 5){

                    clearInterval(intervalo);

                    //Termina el intervalo.

                }

            };

            //TimeOut

            //setTimeout(saludo,3000);

            //Envia un mensaje despues de 3s de iniciar la pagina.

            //Intevarlo

            //var intervalo = setInterval(saludo,1000);

            //Envia un mensaje cada 1s.

***Fechas y tiempo.***

//Variables

            var fecha = new Date();

            //Obtiene la fecha del cliente. (Dia, Hora , Formato).

            fecha.getHours();

            //Obtiene la hora

            console.log(fecha.getHours()-12);

            //muestra en consola la hora

            console.log(fecha.getMinutes());

            //muestra en consola los minutos

            console.log(fecha.getSeconds());

            //muestra en consola los segundos

            console.log(fecha.getDay());

            //muestra en consola el dia de la semana

            console.log(fecha.getDate());

            //muestra en consola el nombre del dia

            console.log(fecha.getMonth());

            //muestra en consola el mes

            console.log(fecha.getFullYear());

            //muestra en consola el año

CSS GRID

***Creacion de css grid***

.contenedor{

Display: grid;

Grids-template-columns : 1fr 1fr 1fr;

A cada columna le asigna una parte del tamaño de forma equivalente.

Grids-template-rows: 150px 300px;

A cada fila le asigna el tamaño que fue asignado y si existen mas columnas le pone uno por default.

Grids-grap: 20px;

Separacion entre filas y columnas.

}

***Medidas minimas y máximas***

.contenido{

Grids-template-rows: minmax(120px , auto) minmax(150,350px) 1fr 1fr;

Agrega un minimo y máximo de tamaño a un contenido mediante minmax.

}

***Funcion Repeat();***

.contenido{

Grid-template-rows: repeat(6, 1fr);

La function repeat hace que se repita una cantidad de veces el número que deseamos poner de filas y cada una de ellas cuanto tendrá de tamaño.

}

***Posicionamiento***

    /\* grid-column-start: 1; \*/

    /\* El div iniciara desde la posicion 1 de las columnas\*/

    /\* grid-column-end: 3; \*/

    /\* El div tomara desde la posicion 1 hasta la posicion final \*/

    /\* grid-row-start: 1; \*/

    /\* El div iniciara desde la posicion 2 de las filas \*/

    /\* grid-row-end: -1; \*/

    /\* El div tomara desde la posicion 2 hasta la posicion final \*/

    /\* grid-column: 1 / 4;

    grid-row: 1 / -1; \*/

    /\* Forma distinta de hacer lo mismo\*/

    /\* grid-area: 1 / 1 / -1 / 2; \*/

    /\* Inicio de fila / Inicio de columna / Fin de la fila / Fin de la columa\*/

    /\* Forma distinta de hacer lo mismo\*/

    grid-column: 1/ span 3;

    /\* la palabra span sirve para indicar las filas/columnas que ocupara el contenido o div \*/

***Grid Areas***

   grid-template-areas: "header    header      header"

                         "sidebar-1 contenido   sidebar-2"

                         "sidebar-1 widgets     sidebar-2"

                         "footer    footer      footer";

    /\*Estructura que tendra la pagina\*/

   background: #eff2f3;

   grid-area: header;

    /\* espeficificamos con grid area: (palabra clave) que columnas y filas tomara en est e caso sera toda la fila 1 \*/

***Alineacion de elementos***

.contenedor{

    display: grid;

    grid-gap: 20px;

    grid-template-columns: repeat(3,1fr);

    grid-template-rows:  repeat(3,200px);

    justify-items: stretch; /\*modifica los lados\*/

    align-items: stretch; /\*modifica hacia arriba/abajo\*/

     /\*  start: alinea la información al inicio del div

        end: alinea la información al final del div

        center: centra la información en el div

        auto: ocupa el mayor espacio disponible

        stretch: ocupa el mayor espacio disponible \*/

}

.contenedor .item{

    justify-self: stretch;

    align-self: stretch;

/\* realiza lo mismo, pero esto se usa para modificar una etiqueta\*/

}

***Alineacion de Tracks***

.contenedor{

justify-content: space-evenly;

align-content: space-evenly;

 /\*  normal,start: alinea todo el contenido al inicio

 center: alinea todo el contenido al centro

 end: alinea todo el contenido al final

 space-around: divide el espacio disponible

 y deja un poco al inicio y al final

space-between: divide el espacio disponible

pero quita el espacio entre el inicio y al final

space-evenly: pone el mismo espaciado

 en inicio fin y entre los contenidos\*/

}

***GRID implícita***

  grid-auto-rows: 100px;

    /\* Sirve para poder darle un valor default al contenido que no cabe en el grid \*/

    grid-auto-flow: dense;

    /\* column,row:Agrega el contenido extra  en forma de columnas o en filas

       dense: Arega los huecos que queden poniendo contenido debajo que tenga el mismo tamaño de hueco\*/

    grid-auto-columns: 100px;

***Auto fill y Auto fit***

    grid-template-columns: repeat(auto-fit,minmax(100px,1fr));

    /\* Auto fit permite es rellenar todo el contenedor con todas las columnas que tenemos disponibles \*/

    /\* Auto fill va a rellenar toda nuestra fila con todas las columnas que  pueda poner dentro y va a crear culumnas implicitas en caso de que se pueda despues cuando no pueda los va a poner en una segunda fila \*/

***Subgrid***

    grid-row: span 3;

   grid-template-rows: auto 1fr auto;

   grid-template-rows:subgrid;

    /\* subgrid : esta funcion mete dentro de un grid anterior o tro grid donde podemos hacer que la informacion quede de forma simétrica o en su defecto si el navegador no acepta este método podemos agregarlo como en la línea anterior \*/

***Pagina web responsive***

@media screen and (max-width:768px){

    .contenedor{

        grid-template-areas: "header        header      header"

                             "contenido     contenido   contenido"

                             "sidebar       sidebar     sidebar"

                             "widget1       widget1     widget2"

                             "footer        footer      footer";

    }

}

/\*Al llegar a limite de tamaño el sitio se volverá responsive\*/

Bootstrap