Mano sosteniendo un cuchillo sobre una mesa de madera

Descripción generada automáticamente con confianza media

4. DESARROLLO DE WEBSITES

**Temas**

Html5

Diseño responsivo

Bootstrap

Conectar a APIs / graph

jQuery o Javascript

Experiencia de usuario

Wordpress

Instructor

**Isaí Fararoni Ramírez**

Ciudad de México 7 de agosto de 2023.

Contenido

[Contenido 2](#_Toc142850432)

[Introducción 5](#_Toc142850433)

[Objetivos del curso 5](#_Toc142850434)

[Herramientas y entorno necesario para empezar 5](#_Toc142850435)

[1. HTML 5 6](#_Toc142850436)

[Introducción 6](#_Toc142850437)

[Etiquetas HTML 6](#_Toc142850438)

[Estructura básica de una página web en HTML 9](#_Toc142850439)

[Etiquetas para formato de texto 13](#_Toc142850440)

[Listas ordenadas, no ordenadas y listas anidadas 17](#_Toc142850441)

[Listas ordenadas 17](#_Toc142850442)

[Listas no ordenadas 18](#_Toc142850443)

[Listas Anidadas 18](#_Toc142850444)

[Imágenes <img> 19](#_Toc142850445)

[Tablas HTML 20](#_Toc142850446)

[Formularios 24](#_Toc142850447)

[Listado de nuevos elementos semánticos 32](#_Toc142850448)

[Ejercicio completo 32](#_Toc142850449)

[Múltiples páginas 32](#_Toc142850450)

[Enlaces con HTML 33](#_Toc142850451)

[Proyecto del alumno 34](#_Toc142850452)

[2. Diseño responsivo 34](#_Toc142850453)

[Introducción 34](#_Toc142850454)

[Conceptos básicos de CSS3 35](#_Toc142850455)

[Agregar estilos a un archivo HTML (Ejercicio) 36](#_Toc142850456)

[Selectores 41](#_Toc142850457)

[Pseudoselectores 43](#_Toc142850458)

[Pseudo-classes 45](#_Toc142850459)

[:right 45](#_Toc142850460)

[:root 45](#_Toc142850461)

[:scope (en-US) 45](#_Toc142850462)

[:target 45](#_Toc142850463)

[:valid 45](#_Toc142850464)

[:visited 45](#_Toc142850465)

[Estructura de una regla CSS 45](#_Toc142850466)

[Modelo de cajas y posicionamiento de elementos 50](#_Toc142850467)

[Maquetación 56](#_Toc142850468)

[Maquetación con porcentajes 57](#_Toc142850469)

[Media queries 61](#_Toc142850470)

[Viewport 63](#_Toc142850471)

[Proyecto del alumno 64](#_Toc142850472)

[3. Bootstrap 64](#_Toc142850473)

[Introducción 65](#_Toc142850474)

[Proyecto del alumno 65](#_Toc142850475)

[4. jQuery o Javascript 65](#_Toc142850476)

[Introducción 65](#_Toc142850477)

[Proyecto del alumno 66](#_Toc142850478)

[5. Conectar a APIs 66](#_Toc142850479)

[Introducción 67](#_Toc142850480)

[Proyecto del alumno 68](#_Toc142850481)

[6. Experiencia de usuario 68](#_Toc142850482)

[Principio de Proximidad 68](#_Toc142850483)

[Principio de la Alineación 69](#_Toc142850484)

[Principio de la Repetición 71](#_Toc142850485)

[Principio de Contraste 73](#_Toc142850486)

[Cuadricula 74](#_Toc142850487)

[Jerarquía Visual 77](#_Toc142850488)

[Combinación de colores 79](#_Toc142850489)

[Tipografía 80](#_Toc142850490)

[7. Wordpress 80](#_Toc142850491)

[Introducción 80](#_Toc142850492)

[Proyecto del alumno – Landing page 81](#_Toc142850493)

[Instalación 83](#_Toc142850494)

[Interfaz 84](#_Toc142850495)

Imagen que contiene Logotipo

Descripción generada automáticamenteIntroducción

En el mundo digital actual, tener la capacidad de crear sitios web atractivos y funcionales es una habilidad de comunicación extremadamente valiosa. En este curso, aprenderás los conceptos básicos y prácticos para crear sitios web funcionales y efectivos para aportar institucionales de tu institución.

HTML es un lenguaje de marcado que se utiliza para crear la estructura de un sitio web. CSS es un lenguaje de hojas de estilo que se utiliza para controlar el aspecto de un sitio web. WordPress es un sistema de gestión de contenido (CMS) que se utiliza para crear y gestionar sitios web.

## Objetivos del curso

Al final de este curso, serás capaz de:

* Crear páginas web básicas usando HTML
* Insertar texto, imágenes, enlaces y otros elementos en tus páginas web
* Aplicar estilos a tus páginas web
* Crear sitios web responsivos
* Aplicar los principios del diseño para crear sitios Web para mejorar la experiencia de tus usuarios
* Crear un sitio web usando WordPress
* Añadir contenido a tu sitio web usando WordPress
* Personalizar el diseño de tu sitio web construido en WordPress
* Consultar y presentar información desde otros sistemas mediante servicios API/REST
* Recolectar información de tus visitantes mediante formularios

**Recursos para el curso**

Los archivos, recursos, ejercicios y solución estarán disponibles en el siguiente repositorio Git <https://github.com/fararoni/curso.websites> .

## Herramientas y entorno necesario para empezar

Ejercicio:

Hacer un recorrido por las herramientas necesarias para empezar en este curso:

* Visual Studio Code para la escritura de las Páginas Web con HTML.
* XAMPP, la herramienta que proporciona un servidor Web para publicar localmente las páginas Web a Desarrollar.
* Navegador Web Chrome, en este curso lo estarás usando como una herramienta de desarrollo.
* PicPick (https://picpick.app/es/), herramienta auxiliar de diseño. Es opcional si las políticas de uso de software lo permiten.

**Nota:**

* Localizar el directorio de trabajo, este se debe ubicar en*: c:\xampp\htdocs*
* Crear una carpeta con el nombre ***curso*** en dicho directorio.

1. HTML 5

## Introducción

**HTML, o HyperText Markup Language**, es un lenguaje de marcado estándar para crear páginas web. Es un lenguaje de texto simple que utiliza etiquetas para describir la estructura de una página web. Estas etiquetas se utilizan para indicar al navegador web cómo mostrar el contenido de la página.

HTML es un lenguaje relativamente sencillo de aprender, y es una habilidad esencial para cualquier persona que quiera crear páginas web[[1]](#footnote-1).

## Etiquetas HTML

Las etiquetas HTML son palabras clave que se utilizan para marcar el contenido de una página web. Las etiquetas HTML se escriben entre los símbolos < y **>.** Por ejemplo, la etiqueta para un encabezado es <h1>.

Las etiquetas HTML pueden ser abiertas o cerradas. Las etiquetas abiertas se utilizan para empezar un **elemento** HTML, y las etiquetas cerradas se utilizan para terminar un elemento HTML. Por ejemplo, la etiqueta para abrir un encabezado es <h1>, y la etiqueta cerrada para un encabezado es </h1>. El nombre de un elemento HTML es el nombre utilizado entre paréntesis angulares.

Anatomía de un elemento HTML

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Fuente: <https://developer.mozilla.org/es/docs/Learn/HTML/Introduction_to_HTML/Getting_started/grumpy-cat-small.png>

Las principales partes de un elemento[[2]](#footnote-2) son:

* La etiqueta de apertura: consiste en el nombre del elemento (en este caso, p), encerrado entre paréntesis angulares de apertura y cierre. Esta etiqueta de apertura marca dónde comienza el elemento o comienza a tener efecto. En este ejemplo, precede al comienzo del texto del párrafo.
* El contenido: Este es el contenido del elemento. En este ejemplo, es el texto del párrafo.
* La etiqueta de cierre: Es lo mismo que la etiqueta de apertura, excepto que incluye una barra diagonal antes del nombre del elemento. Esto indica dónde termina el elemento; en este caso, dónde finaliza el párrafo. No incluir una etiqueta de cierre es un error común de principiante, y puede conducir a extraños resultados.

El elemento lo conforman la etiqueta de apertura, seguida del contenido, seguido de la etiqueta de cierre.

Elementos anidados

Se pueden poner elementos dentro de otros elementos. Esto se llama anidamiento. Los elementos tienen que abrirse y cerrarse en el orden correcto para que estén claramente dentro o fuera el uno del otro.

Correcto:

**<p>** Mi nombre es **<strong>** [Tu Nombre] **</strong>,** y esta es mi página web **</p>**

Incorrecto:

**<p>** Mi nombre es **<strong>** [Tu Nombre] **</p>,** y esta es mi página web **</strong>**

Analice la estructura del siguiente código:

**<p>**

Mi nombre es

**<strong>** [Tu Nombre]

**</strong>,**

y esta es mi página web

**</p>**

Elementos de bloque (block) y elementos en línea (inline)

Hay dos categorías importantes de elementos en HTML

* **Los elementos de bloque** forman un bloque visible en la página. Aparecerán en una línea nueva después de cualquier contenido anterior. Cualquier contenido que vaya después también aparecerá en una línea nueva. Los elementos a nivel de bloque suelen ser elementos estructurales de la página. Por ejemplo, un elemento a nivel de bloque puede representar encabezados, párrafos, listas, menús de navegación o pies de página. Un elemento a nivel de bloque no estaría anidado dentro de un elemento en línea, pero podría estar anidado dentro de otro elemento a nivel de bloque.
* **Los elementos en línea** están contenidos dentro de elementos de bloque y delimitan solo pequeñas partes del contenido del documento; (no párrafos enteros o agrupaciones de contenido) **Un elemento en línea no hará que aparezca una nueva línea en el documento**. Suele utilizarse con texto. Por ejemplo es el caso de un elemento <a> (hipervínculo) o elementos de énfasis como <em> o <strong>.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Fuente: https://www.csssolid.com/css-position.html

Elementos vacíos

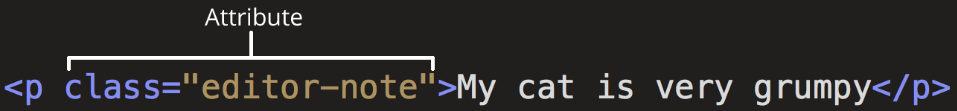
Algunos elementos consisten solo en una etiqueta única (sin etiqueta de cierre), que se utiliza generalmente para insertar/incrustar algo en el documento en el lugar donde se le quiere incluir. Por ejemplo, el elemento <img> inserta una imagen en la página:

Ejemplo:

<img src=" https://web.diputados.gob.mx/assets/images/logo.png" />

Atributos

Los elementos también pueden tener atributos. Los atributos tienen esta estructura:



Un atributo se escribe:

* Un espacio entre este y el nombre del elemento. (Para un elemento con más de un atributo, los atributos también deben estar separados por espacios).
* El nombre del atributo, seguido por un signo igual.
* Un valor del atributo, rodeado de comillas de apertura y cierre.

Los atributos contienen información extra sobre el elemento que no se mostrará en el contenido.

Atributos booleanos

Los atributos booleanos solo pueden tener un valor, que generalmente es el mismo que el nombre del atributo.

Ejemplo:

<input type="text" disabled="disabled">

## Estructura básica de una página web en HTML

Una página web HTML está estructurada en tres secciones principales:

|  |  |
| --- | --- |
| La cabecera (**HEAD**)  El cuerpo (**BODY**)  El pie (**FOOTER**) |  |

* La **cabecera (head)** contiene información sobre la página web, como el título, la descripción y los metadatos.
* El **cuerpo (body)** contiene el contenido de la página web, como el texto, las imágenes, los enlaces y otros elementos.
* El **pie (footer)** contiene información adicional sobre la página web, como los créditos, los enlaces a las redes sociales y otros elementos.

Ahora una estructura estandarizada que cumpla con requisitos básicos de funcionamiento tiene el siguiente elemento adicional:

* Contenedor. Es todo lo que se encuentra en la regla Body o en alguna división de nuestra estructura. Dentro del contenedor están todos los elementos del sitio; módulos, imágenes, contenidos. Este contenedor puede tener un ancho fijo o adaptativo en el caso de las páginas web responsive.

**Ejercicio.**

1. Abrir Visual Studio Code
2. Abrir la carpeta c:\xampp\htdocs\curso

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

*Verificar que estas en el directorio curso*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Crear el siguiente archivo

|  |
| --- |
|  |

1. Escribe el siguiente código en el archivo creado

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <title>Mi Página Web</title>  </head>  <body>      <header>          <h1>Bienvenido a Mi Página Web</h1>      </header>      <main>          <section>              <h2>Sobre mí</h2>             <p>                Mi nombre es                <strong>                    [Tu Nombre]                </strong>,                Trabajo aquí  <img src=" https://web.diputados.gob.mx/assets/images/logo.png" />             </p>          </section>      </main>      <footer>          <p>Derechos de autor © 2023 - Mi Página Web</p>      </footer>  </body>  </html> |

1. Verifica que se esté ejecutando el servidor Web desde el panel de XAMPP

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Abre el navegador Chrome y entra a la siguiente dirección: <http://localhost/curso/01.estructura.html>

Texto

Descripción generada automáticamente

En este ejercicio, hemos creado una estructura básica para una página web.

Comienza con la declaración **<!DOCTYPE html>,** seguida de la etiqueta **<html>** que envuelve todo el contenido de la página. Dentro de **<html>**, encontramos la etiqueta **<head>** que contiene metadatos, como la codificación de caracteres, el título de la página y puede tener un enlace a una hoja de estilos externa.

El contenido principal de la página se encuentra dentro de la etiqueta <body>, donde tenemos el encabezado (<header>), el contenido principal (<main>), y el pie de página (<footer>). Dentro del encabezado, tenemos un título (<h1>).

En la sección de contenido principal, hay una sección (<section>) con subtítulos (<h2>) y párrafos (<p>).

Es importante mencionar que este es solo un ejemplo básico y que la estructura de una página web puede variar según los requisitos y el contenido específico que desees incluir.

**<!DOCTYPE html>**

Es una etiqueta que informa al navegador qué versión de HTML (o XML) se usó para escribir el documento. podemos referirnos a ella como "document type declaration" o por las siglas "DTD".

**<meta charset="UTF-8">**

En HTML5 el juego de caracteres a usar es siempre UTF-8.

El editor que se usa para escribir las páginas también lee y guarda en UTF8, puede verificar en Visual Studio Code <ctrl>+<,> files encoding.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Etiquetas para formato de texto

La mayor parte del texto estructurado está compuesto por encabezados y párrafos

Encabezados

Hay seis elementos de encabezado: <h1>, <h2>, <h3>, <h4>, <h5> y <h6>. Cada elemento representa un nivel de contenido diferente en el documento; <h1> representa el título principal, <h2> representa el subtítulo, <h3> representa el subtítulo del subtítulo, y así sucesivamente.

Ejemplo:

<h1>Título 1</h1>

<h2>Título 2</h2>

<h3>Título 3</h3>

<h4>Título 4</h4>

<h5>Título 5</h5>

<h6>Título 6</h6>

Párrafos

Cada párrafo tiene que estar delimitado por un elemento <p>

Ejemplo:

<p>Soy un párrafo, ¡desde luego que lo soy!</p>

Negrita

El elemento <**strong>** marca con especial énfasis las partes más importantes de un texto.

Ejemplo:

<p>

<em>El dinero</em> es importante pero <strong>la salud</strong> lo es más.

</p>

**em**. **em** sirve para dar énfasis y **strong** para dar mucho énfasis

Cursiva

El elemento <i> Muestra el texto marcado con un estilo en cursiva o itálica.

Ejemplo:

<p>

Texto normal y... <i>Texto 'inclinado'</i>

</p>

Cita

HTML dispone de dos elementos para el marcado de citas; puedes marcar la cita como un bloque o como un elemento en línea.

**Cita como bloque <blockquote>**

Si una sección de contenido a nivel de bloque (ya sea un párrafo, varios párrafos, una lista, etc.) se cita en otro lugar, debes envolverla dentro de un elemento <blockquote> para indicarlo, e incluye una URL que apunte a la fuente de la cita dentro de un atributo cite.

Ejemplo:

<p>A continuación se muestra una cita en bloque independiente...</p>

<blockquote

cite="https://comunicacionsocial.diputados.gob.mx/revista/index.php/pluralidad/elecciones-2024-y-la-inteligencia-artificial-\_20230728\_143945">

<h1>

Elecciones 2024 y la <em>inteligencia artificial<em></h1>

<p>La tecnología avanza a pasos agigantados</>

</blockquote>

Cita en línea <q>

Ejemplo:

<p>A continuación se muestra una cita en línea...</p>

<q

cite="https://comunicacionsocial.diputados.gob.mx/revista/index.php/pluralidad/elecciones-2024-y-la-inteligencia-artificial-\_20230728\_143945">

Elecciones 2024 y la <em>inteligencia artificial<em>

La tecnología avanza a pasos agigantados.

</q>

Saltos de línea

El elemento HTML line break <br> produce un salto de línea en el texto (retorno de carro). Es útil donde la división de las líneas es significante. No utilices <br> para incrementar el espacio entre líneas de texto; para ello utiliza la propiedad margin de CSS o el elemento <p>.

Separador

El elemento HTML <hr> representa un cambio de tema entre párrafos (por ejemplo, un cambio de escena en una historia, un cambio de tema en una sección). En versiones previas de HTML representaba una línea horizontal. Aún puede ser representada como una línea horizontal en los navegadores visuales, pero ahora es definida en términos semánticos y no tanto en términos representativos.

Ejemplo:

<p>Este es un párrafo únicamente visual para ver el uso de la etiqueta hr</p>

<hr/>

<p>Este es el segundo párrafo, separado del primero por la etiqueta hr, que puede representarse mediante una línea horizontal.</p>

Contenedor en línea <span>

span - abarcar. Es un contenedor en línea. Sirve para aplicar estilo al texto o agrupar elementos en línea.

Sus etiquetas son: <span> y </span> (ambas obligatorias).

Ejemplo:

<div style="border: 1px dotted blue;">

<h4>Ejemplo de div y span</h4>

<p>

Esto es un párrafo dentro de un div,

<span style="color: red;"> y esto un span dentro de un párrafo.

</span>

</p>

</div>

Por definición span no debe tener ningún estilo predefinido. Estos dos párrafos deben verse igual:

<p> Esto es un párrafo, bla, bla, bla. </p>

<p> Esto es un párrafo, <span>bla, bla, bla.</span> </p>

Contenedor en bloque <div>

div de "division" -división . Sirve para crear secciones o agrupar contenidos.

Sus etiquetas son: <div> y </div> (ambas obligatorias).

**Ejercicio:**

1. Escribe el siguiente código en un archivo con el nombre 02.texto.html:

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <title>Mi Página Web</title>

</head>

<body>

    <header>

        <h1>Elecciones 2024 y la inteligencia artificial</h1>

        <p><strong>Luis Enrique Lima Colin</strong> / <i>Director Nacional de Formación y Capacitación de Juventud

                Real</i></p>

        <hr>

    </header>

    <main>

        <section>

            <h2>Avance Tecnológico</h2>

            <p>La tecnología avanza a pasos agigantados y aunque mucho pareciera sacado de un li-<br />bro o película de

                <strong>ciencia ficción</strong>.

            </p>

            <p>Para los fines de este artículo nos centraremos en este último aspecto.

            </p>

        </section>

        <section>

            <h2>ChatGPT</h2>

            <h3>Tipos de aplicaciones</h3>

            <p>Es de destacar que a partir de la puesta en funcionamiento de chat gpt al público</p>

            <h4>Desarrollo de presentaciones</h4>

            <blockquote>“Redacta un discurso para convencer a los votantes de 60 años de edad que soy la me-jor opción como candidato para diputado”</blockquote>

            <h4>Resúmenes de libros</h4>

            <h4>Desarrollo y creación de logotipos</h4>

        </section>

    </main>

    <footer>

        <hr>

        <p>Derechos de autor © 2023 - Mi Página Web</p>

    </footer>

</body>

</html>

El HTML se usa únicamente para definir el contenido. Los atributos de formato no le corresponden al HTML porque están definiendo la forma con la que un párrafo debe de representarse, su estilo, y no el contenido. Estos aspectos se deben hacer mediante el lenguaje CSS, que sirve para definir el estilo, la forma.

## Listas ordenadas, no ordenadas y listas anidadas

En HTML hay tres diferentes tipos ordenadas, no ordenadas y listas anidadas:

## Listas ordenadas

Las listas ordenadas son aquellas en las que el orden de los elementos sí importa. Tomemos como ejemplo una lista de instrucciones para seguir un itinerario:

* Cada lista ordenada comienza con un elemento <ol> («ordered list») que delimita todos los elementos de la lista.
* El siguiente paso es delimitar cada artículo de la lista con un elemento <li> («list item»).

Ejemplo

<ol>

<li>Conduce hasta el final de la calle</li>

<li>Gira a la derecha</li>

<li>Sigue derecho por las dos primeras glorietas</li>

<li>Gira a la izquierda en la tercer glorieta</li>

<li>El colegio está a tu derecha, 300 metros más adelante</li>

</ol>

## Listas no ordenadas

Las listas no ordenadas se usan para marcar listas de artículos cuyo orden no es importante. Por ejemplo, una lista de compras:

* Cada lista desordenada comienza con un elemento <ul> («unordered list») que delimita todos los elementos de la lista.
* El siguiente paso es delimitar cada artículo de la lista con un elemento <li> («list item»).

Ejemplo

<ul>

<li>leche</li>

<li>huevos</li>

<li>pan</li>

<li>hummus</li>

</ul>

## Listas Anidadas

Permite tener subelementos bajo elementos de rango superior. Tomemos el siguiente ejemplo:

**<ol>**

<li>Pela el ajo y picalo en trozos gruesos.</li>

<li>Retira las semillas, y cortarlo en trozos gruesos.</li>

<li>Mete todos los alimentos en un procesador de alimentos.</li>

<li>Procesa todos los ingredientes hasta conseguir una pasta.

**<ul>**

<li>Si deseas un hummus "grueso", procésalo corto tiempo.</li>

<li>Pica más tiempo si se desea obtener un hummus "suave".</li>

**</ul>**

</li>

**</ol>**

**Ejercicio. 03.listas.html**

## Imágenes <img>

Preparación:

1. Crear la carpeta **imagemes** en el directorio de trabajo:

Diagrama

Descripción generada automáticamente

1. Descarga las siguientes imagenes y colócalas en el directorio imágenes.

El elemento de imagen HTML **<img>** muestra una imagen en el documento. Los navegadores no siempre muestran la imagen a la que el elemento hace referencia. Por accesibilidad se debe usar un texto en el atributo alt.

Este elemento incluye atributos globales.

align (obsoleto) Alineamiento de la texto respecto al contexto que la rodea.

alt Este atributo define el texto alternativo que describe la imagen, texto que los usuarios verán si la URL de la imagen es errónea o la imagen tiene un formato no soportado o si la imagen aún no se ha descargado.

border (obsoleto) Anchura del borde alrededor de la imagen.

height La altura de la imagen en píxeles CSS en HTML5 o píxeles o como porcentaje en HTML4.

**src** La URL de la imagen. Este atributo es obligatorio para el elemento <img>.

width El ancho de la imagen en píxeles CSS en HTML5, o píxeles o porcentaje en HTML4.

Formatos de imagen soportada:

JPEG

GIF, including animated GIFs

PNG

APNG

SVG

BMP

BMP ICO

PNG ICO

Ejemplo:

<img src="mdn-logo-sm.png" alt="MDN" />

<a href="https://developer.mozilla.org/"

><img src="mdn-logo-sm.png" alt="MDN"

/></a>

## Tablas HTML

El Elemento de Tabla HTML (<table>) representa datos en dos o más dimensiones.

*Antes de la creación de CSS, los elementos HTML <table> eran utilizados frecuentemente para la disposición (posicionamiento) de una página. Este uso ha sido descontinuado desde HTML 4, y el elemento no debe ser usado para dichos fines.*

Estructura de elementos permitidos

**En este orden:**

* un elemento opcional <caption>,
* cero o mas elementos <colgroup>,
* un elemento opcional <thead> ,
* una de las dos alternativas:
  + un elemento <tfoot> , seguido por:
    - cero o mas elementos <tbody> (en-US),
    - o uno o mas elementos <tr>,
  + una segunda alternativa seguida por un elemento opcional <tfoot>:
    - cualquier cero o mas elementos <tbody> (en-US),
    - uno o mas elementos <tr>

Ejemplos:

<table>

<tr>

<td>John</td>

<td>Doe</td>

</tr>

<tr>

<td>Jane</td>

<td>Doe</td>

</tr>

</table>

**Ejemplo 2**

<table>

<tr>

<th>First name</th>

<th>Last name</th>

</tr>

<tr>

<td>John</td>

<td>Doe</td>

</tr>

<tr>

<td>Jane</td>

<td>Doe</td>

</tr>

</table>

**Ejemplo 3.**

<table>

<thead>

<tr>

<th>Header content 1</th>

<th>Header content 2</th>

</tr>

</thead>

<tfoot>

<tr>

<td>Footer content 1</td>

<td>Footer content 2</td>

</tr>

</tfoot>

<tbody>

<tr>

<td>Body content 1</td>

<td>Body content 2</td>

</tr>

</tbody>

</table>

**Ejemplo 4:**

<table>

<colgroup span="4" class="columns"></colgroup>

<tr>

<th>Countries</th>

<th>Capitals</th>

<th>Population</th>

<th>Language</th>

</tr>

<tr>

<td>USA</td>

<td>Washington D.C.</td>

<td>309 million</td>

<td>English</td>

</tr>

<tr>

<td>Sweden</td>

<td>Stockholm</td>

<td>9 million</td>

<td>Swedish</td>

</tr>

</table>

Ejemplo 5:

<table>

<colgroup>

<col class="column1" />

<col class="columns2plus3" span="2" />

</colgroup>

<tr>

<th>Lime</th>

<th>Lemon</th>

<th>Orange</th>

</tr>

<tr>

<td>Green</td>

<td>Yellow</td>

<td>Orange</td>

</tr>

</table>

**Ejemplo 6**

<table>

<caption>

Awesome caption

</caption>

<tr>

<td>Awesome data</td>

</tr>

</table>

**Ejercicio:**

1. Crear un archivo 04.tablas.html
2. Integra todos los ejemplos de tabla en el archivo html
3. Visualiza en el navegador.

## Formularios

El elemento HTML form (<form>) representa una sección de un documento que contiene controles interactivos que permiten a un usuario enviar información a un servidor web.

Atributos

Como cualquier otro elemento HTML, este elemento soporta atributos globales.

**Action.** La URI de un programa que procesa la información enviada por medio del formulario. Este valor puede ser sobreescrito por un atributo formaction en un <button> o en el elemento<input>.

**enctype**

Cuando el valor del atributo method es post, este atributo es el tipo MIME del contenido que es usado para enviar el formulario al servidor. Los posibles valores son:

* application/x-www-form-urlencoded: El valor por defecto si un atributo no está especificado.
* multipart/form-data: Usar este valor si se está usando el elemento <input> con el atributo type ajustado a "file".
* text/plain (HTML5)

Este valor puede ser sobreescrito por un atributoformenctypeen un <button> o un elemento <input>.

**method**

El método HTTP que el navegador usa para enviar el formulario. Valores posibles son:

* **post**: Corresponde al método POST HTTP ; los datos del formulario son incluidos en el cuerpo del formulario y son enviados al servidor.
* **get**: Corresponde al método GET HTTP; los datos del formulario son adjuntados a la URI del atributo action , con un '?' como separador, y la URI resultante es enviada al servidor. Use este método cuando el formulario no tiene efectos secundarios y contiene solo caracteres ASCII.

Este valor puede ser sobreescrito por un atributo formmethod en un <button> o elemento <input>.

**name**

El nombre del formulario. En HTML4 ha quedado en desuso (debe usarse un id en su lugar). Debe ser único entre los formularios en un documento y no una cadena vacia en HTML5.

**target**

Un nombre o keyword indicando donde mostrar la respuesta que es recibida después de enviar el formulario.

**Ejemplo:**

<!-- Formulario simple que enviará una petición GET -->

<form action="">

<label for="GET-name">Nombre:</label>

<input id="GET-name" type="text" name="name" />

<input type="submit" value="Save" />

</form>

<!-- Formulario simple que enviará una petición POST -->

<form action="" method="post">

<label for="POST-name">Nombre:</label>

<input id="POST-name" type="text" name="name" />

<input type="submit" value="Save" />

</form>

<!-- Formulario con conjunto de campos, leyenda y etiqueta -->

<form action="" method="post">

<fieldset>

<legend>Título</legend>

<input type="radio" name="radio" id="radio" />

<label for="radio">Clic aquí</label>

</fieldset>

</form>

<Input>

El elemento HTML <input> se usa para crear controles interactivos para formularios basados en la web con el fin de recibir datos del usuario.Hay disponible una amplia variedad de tipos de datos de entrada y widgets de control.

Atributo global:

**type.** El tipo de control a mostrar. Su valor predeterminado es text, si no se especifica este atributo. Los posibles valores son:

* button: Botón sin un comportamiento específico.
* checkbox: Casilla de selección. Se debe usar el atributo value para definir el valor que se enviará por este elemento. Se usa el atributo checked para indicar si el elemento está seleccionado. También se puede usar el atributo indeterminate (el cual solo se puede establecer programaticamente) para indicar que la casilla está en un estado indeterminado (en la mayoría de las plataformas, se dibuja una línea horizontal a través de la casilla).
* color: Control para espicificar un color. Una interfaz de selección de color no requiere más funcionalidad que la de aceptar colores simples como texto (más información).
* date: Control para introducir una fecha (año, mes y día, sin tiempo).
* datetime: Obsoleto Control para introducir una fecha y hora (horas, minutos, segundos y fracción de segundo), basado en la zona horaria UTC. Esta característica ha sido removida de WHATWG HTML.
* datetime-local: Control para introducir fecha y hora, sin zona horaria específica.
* email: Campo para introducir una dirección de correo electrónico. El valor introducido se valida para que contenga una cadena vacía o una dirección de correo válida antes de enviarse. Las pseudo-clases :valid y :invalid son aplicadas segun corresponda. Si se agrega el atributo **multiple** se pueden ingresar varias direccione de correo electrónico.
* file: Control que permite al usuario seleccionar un archivo. Se puede usar el atributo accept para definir los tipos de archivo que el control podrá seleccionar.
* hidden: Control que no es mostrado en pantalla, pero cuyo valor es enviado al servidor.
* image: Botón de envío de formulario con gráfico. Se debe usar el atributo src para definir el origen de la imagen y el atributo alt para definir un texto alternativo. Se puede usar los atributos height y width para definir el tamaño de la imagen en píxeles.
* month: Control para introducir un mes y año, sin zona horaria específica.
* number: Control para introducir un número de punto flotante.
* password: Control de línea simple cuyo valor permanece oculto. Se puede usar el atributo maxlength para especificar la longitud máxima del valor que se puede introducir.
* radio: Botón radio. Se debe usar el atributo value para definir el valor que se enviará por este elemento. Se usa el atributo checked para indicar si el elemento está seleccionado de forma predeterminada. Los botones radio que tengan el mismo valor para su atributo name están dentro del mismo "grupo de botones radio". Solo un botón radio dentro de un grupo puede ser seleccionado a la vez.
* range: Control para introducir un número cuyo valor exacto no es importante. Este control usa los siguientes valores predeterminados si no se especifica cada atributo:
  + min: 0
  + max: 100
  + value: min + (max - min) / 2, o min si max es menor que min
  + step: 1
* reset: Botón que restaura los contenidos de un formulario a sus valores predeterminados.
* search: Cuadro de texto de línea simple para introducir textos de búsqueda. Los saltos de línea son eliminados automáticamente del valor introducido.
* submit: Botón que envía el formulario.
* tel: Control para introducir un número telefónico. Los saltos de línea son eliminados automáticamente del valor introducido, pero no hay otra sintaxis forzada. Se pueden usar atributos como pattern y maxlength para restringir los valores introducidos en este control. Las pseudo-clases CSS :valid y :invalid son aplicadas segun corresponda.
* text: Campo de texto de línea simple. Los saltos de línea son eliminados automáticamente del valor introducido.
* time: Control para introducir un valor de tiempo sin zona horaria específica.
* url: Campo para editar una URL. El valor introducido se valida para que contenga una cadena vacía o una ruta URL absoluta antes de enviarse. Los saltos de línea y espacios en blanco al principio o al final del valor son eliminados automáticamente. Se pueden usar atributos como pattern y maxlength para restringir los valores introducidos en el control. Las pseudo-clases :valid y :invalid son aplicadas segun corresponda.
* week: Control para introducir una fecha que consiste en número de semana del año y número de semana sin zona horaria específica.

Atributos del elemento input

* + **id.**
  + Required:
  + Autocapitalize: Controla si se deben cambiar los textos a mayúsculas automáticamente mientras son introducidos/editados por el usuario. Valores disponibles: none, sentences, words, characters
  + Autofocus: Este atributo Booleano permite especificar que un control de formulario tenga el cursor cuando la página se carga.
  + Checked. Atributo booleano que indica que la opción se encuentra seleccionada, solo aplica para radio y checkbox.
  + Disabled : Indica que el control no está disponible para interacción, el evento click no estará disponible. Su información no se envía en el formulario.
  + Form: Permite poner elementos <input> en cualquier parte dentro de un documento y asociarlos a un formulario por medio del atributo id del formulario.
  + Inputmode. Una pista para el navegador sobre qué teclado mostrar: verbatim: Alfanumérico; numeric: Escritura en caracteres numéricos; tel: Escritura de teléfonos; email: Escritura de correo electrónico; url: Escritura de URL. Es preferible el uso de <input type="url"> en lugar de este atributo.
  + List: Es el identificador de <datalits> con las opciones de ayuda a los usuarios para seleccionar valores válidos de manera más rápida y precisa.
  + Max, Min: El valor máximo y mínimo (numérico o fecha-hora) para este elemento, max no debe ser menor que su valor mínimo (atributo min).
    - Step. Trabaja con los atributos min y max, para limitar el incremento de valores numéricos o de fecha-hora.
  + Maxlength. Solo aplica para text, email, search, password, tel, o url, este atributo especifica el número máximo de caracteres.
  + Minlength. Solo aplica para text, email, search, password, tel, o url, este atributo especifica la longitud mínima de caracteres.
  + Name. El nombre del control, el cual es enviado con los datos del formulario.
  + Pattern: Una expresión regular contra la que es verificado el valor del control, que puede ser usada con valores de type de text, tel, search, url y email.
  + Placeholder. Una pista para el usuario sobre lo que puede introducir en el control. El texto no debe contener saltos de línea.

No se recomienda usar el atributo placeholder en lugar de un elemento <label>, pues sus propósitos son diferentes. El elemento <label> describe el rol del elemento en el formulario (es decir, indica qué tipo de información se espera), y el atributo placeholder es una pista sobre el formato que debe tener el contenido.

* + Readonly. Modifica el control para que el usuario no puede su valor.
  + Required. Este atributo especifica que el usuario debe llenar el control antes de enviarlo al formulario.
  + Size. El tamaño inicial del control. Este valor es en píxeles, a menos que el atributo type sea text o password, en cuyo caso será el número entero de caracteres. A partir de HTML5, este atributo aplica sólo cuando el atributo type es text, search, tel, url, email, o password.
  + Src. Solo aplica para el elemento con type=”image”, este atributo especifica el URI para la ubicación de la imagen a mostrar en el botón de envío gráfico.
  + Tabindex. La posición del elemento en el orden de navegación por la tecla Tab dentro del documento.
  + Value. El valor inicial del control. Este atributo es opcional, excepto cuando el atributo type es radio o checkbox.

Atributos para todos los elementos HTML

* + Id. El atributo global id define un identificador único (ID) el cual no debe repetirse en todo el documento. Su propósito es identificar el elemento al vincularlo (usando un identificador de fragmento),

en scripts u hojas de estilo (con CSS). El valor de este atributo es una cadena de caracteres opaca: es decir, el autor del sitio no debe usarlo para proporcionar información. De forma particular, en sentido semántico, por ejemplo, no debe ser derivado de la misma cadena. Los elementos pueden tener sólo un ID definido mediante el atributo ID.

Ejemplo:

<p>A common form that includes input tags</p>

<form action="getform.php" method="get">

<label>First name: <input type="text" /></label><br />

<label>Last name: <input type="text" /></label><br />

<label>E-mail: <input type="email" /></label><br />

<input type="submit" value="Submit" />

</form>

<label>

El Elemento HTML <label> representa una etiqueta para un elemento en una interfaz de usuario. Este puede estar asociado con un control ya sea mediante la utilizacion del atributo for, o ubicando el control dentro del elemento label. Tal control es llamado "el control etiquetado" del elemento label.

**Ejemplo:**

<!-- Un simple ejemplo de un label con el atributo for -->

<label for="Name">Click me</label>

<input type="text" id="Name" name="Name" />

<!-- Aun mas simple -->

<label>Click me <input type="text" id="Name" name="Name" /></label>

<button>

La etiqueta de HTML <button> representa un elemento cliqueable de tipo botón que puede ser utilizado en formularios o en cualquier parte de la página que necesite un botón estándar y simple de aplicar.

<select>

El elemento select (<select>) de HTML representa un control que muestra un menú de opciones. Las opciones contenidas en el menú son representadas por elementos <option>, los cuales pueden ser agrupados por elementos <optgroup>. La opcion puede estar preseleccionada por el usuario.

Ejemplo

<!-- The second value will be selected initially -->

<select name="select">

<option value="value1">Value 1</option>

<option value="value2" selected>Value 2</option>

<option value="value3">Value 3</option>

</select>

Etiquetas semánticas

Principio del formulario

<datalist>

La etiqueta <datalist> en HTML5 se utiliza para proporcionar una lista de opciones sugeridas o predefinidas para un elemento de entrada de tipo texto, como <input type="text">. Esta lista de opciones ayuda a los usuarios a seleccionar valores válidos de manera más rápida y precisa, reduciendo la posibilidad de errores de entrada.

Ejemplo:

<label for="pais">País:</label>

<input type="text" id="pais" name="pais" list="paises">

<datalist id="paises">

<option value="Estados Unidos">

<option value="Canadá">

<option value="México">

<option value="Brasil">

<option value="Argentina">

<!-- Agrega más opciones según sea necesario -->

</datalist>

<fieldset>

La etiqueta <fieldset> en HTML5 se utiliza para agrupar y organizar elementos de formulario relacionados. Proporciona un contenedor visual y semántico que ayuda a estructurar los campos de un formulario y a mejorar la accesibilidad para los usuarios y las tecnologías de asistencia, como lectores de pantalla.

El elemento <fieldset> se utiliza comúnmente junto con la etiqueta <legend> para proporcionar un título o una descripción para el grupo de elementos de formulario contenidos dentro del <fieldset>.

Ejemplo:

<fieldset>

<legend>Información Personal</legend>

<label for="nombre">Nombre:</label>

<input type="text" id="nombre" name="nombre">

<br>

<label for="apellido">Apellido:</label>

<input type="text" id="apellido" name="apellido">

</fieldset>

<fieldset>

<legend>Preferencias</legend>

<input type="checkbox" id="deportes" name="intereses" value="Deportes">

<label for="deportes">Deportes</label>

<br>

<input type="checkbox" id="musica" name="intereses" value="Música">

<label for="musica">Música</label>

</fieldset>

<button type="submit">Enviar</button>

<textarea>

La etiqueta <textarea> en HTML5 se utiliza para crear un campo de texto multilínea en formularios web. A diferencia de los campos de entrada de texto normal (<input type="text">), que son campos de una sola línea, <textarea> permite a los usuarios ingresar y editar texto en varias líneas.

<label for="comentario">Escribe tu comentario:</label>

<textarea id="comentario" name="comentario" rows="4" cols="50"></textarea>

## Listado de nuevos elementos semánticos

Los elementos semánticos agregados a HTML5 son:

* <article>
* <aside>
* <details>
* <figcaption>
* <figure>
* <footer>
* <header>
* <main>
* <mark>
* <nav>
* <section>
* <summary>
* <time>

## Ejercicio completo

Ejercicio

1. Esquematizar una página
2. Construir la página

## Múltiples páginas

Ejercicio

1. Esquematizar un sitio con varias páginas
   1. Inicio.html
   2. acercade.html
   3. contacto.html
2. Construir las paginas

.

## Enlaces con HTML

Los hipervínculos nos permiten vincular documentos a otros documentos o recursos, vincular a partes específicas de documentos o hacer que las aplicaciones estén disponibles en una dirección web.

Un enlace básico se crea incluyendo el texto (o cualquier otro contenido, que queramos convertir en un enlace usando un elemento ancla <a>, dándole un atributo href que contendrá la dirección web hacia dónde queremos que apunte el enlace.

**Ejemplo 1:**

<p>

Crea un enlace a

<a href="https://comunicacionsocial.diputados.gob.mx//">la página de la revista cámara</a>.

</p>

**Ejemplo 2:**

<p>

Crea un enlace a

<a href=<https://comunicacionsocial.diputados.gob.mx//>

title="Revista digital interactiva”

>la página de la revista cámara</a>.

</p>

## Proyecto del alumno

1. Forma, Círculo

   Descripción generada automáticamenteDiseño responsivo

## Introducción

En la era actual de la tecnología, los usuarios acceden a internet desde una variedad de dispositivos, como computadoras de escritorio, laptops, tabletas y teléfonos inteligentes. Como desarrolladores web, es esencial asegurarnos de que nuestras creaciones se adapten perfectamente a todas estas pantallas y ofrezcan una experiencia de usuario fluida y atractiva. Es ahí donde entra en juego el Diseño Responsivo.

El diseño responsivo es importante porque cada vez más personas acceden a Internet a través de dispositivos móviles. Según el Pew Research Center, en 2022, el 81% de los adultos en los Estados Unidos utilizaban Internet, y el 45% lo hacía a través de un teléfono móvil.

Los sitios web que no son responsivos pueden ser difíciles de usar en dispositivos móviles. El texto puede ser demasiado pequeño para leer, los enlaces pueden ser difíciles de hacer clic y los formularios pueden ser difíciles de completar. Esto puede conducir a una mala experiencia del usuario y a una pérdida de clientes potenciales.

El Diseño Responsivo es una metodología de desarrollo web que busca crear sitios que se ajusten y respondan automáticamente al tamaño y resolución del dispositivo utilizado para verlos. Esto se logra mediante el uso de tecnologías como HTML, CSS y JavaScript, y permite que el contenido, las imágenes y los elementos de navegación se reorganicen y se presenten de manera óptima en cada pantalla, sin sacrificar la calidad visual ni la facilidad de uso.

Hay una serie de técnicas que se pueden utilizar para crear sitios web responsivos. Una de las técnicas más comunes es utilizar las hojas de estilo en cascada (CSS) la cual usaremos en este curso y veremos una introducción a continuación.

Otra técnica que se puede utilizar para crear sitios web responsivos es utilizar los media queries. Los media queries son declaraciones que se utilizan para especificar cómo se debe mostrar una página web en función de diferentes dispositivos.

El diseño responsivo es una parte importante del desarrollo web moderno. Es esencial para crear sitios web que se vean bien y sean fáciles de usar en una amplia gama de dispositivos.

Aquí tienes algunos beneficios de utilizar el diseño responsivo en tu sitio web:

* Mejor experiencia del usuario: El diseño responsivo permite que tus visitantes vean tu sitio web en cualquier dispositivo, lo que les proporciona una mejor experiencia.
* Mayor alcance: El diseño responsivo te permite llegar a un público más amplio, ya que las personas acceden a Internet a través de una variedad de dispositivos.
* Mejores resultados de búsqueda: Google y otros motores de búsqueda dan prioridad a los sitios web que son responsivos, lo que puede ayudarte a mejorar tu clasificación en los resultados de búsqueda.
* Menores coLastos: El diseño responsivo puede ayudarte a ahorrar dinero en el desarrollo y el mantenimiento de tu sitio web, ya que no necesitas crear versiones separadas para diferentes dispositivos.

#### Ejercicio

Compara los sitios en la tabla, indica cual te parece responsivo.

1. Abre los sitios en el navegador Chrome.
2. Activa el modo desarrollador presionando la tecla **F12**, a continuación <CTRL><SHIFT><M>, selecciona el modelo IPhoneXR
3. A continuación, ingresa a los sitios mobile-friendly y amiresponsive
4. Califica con un simple si o no.

|  |  |  |  |
| --- | --- | --- | --- |
|  | IPhoneXR  Responsivo  (Persepción) | mobile-friendly | amiresponsive |
| https://www.samsung.com/mx/ |  |  |  |
| https://www.apple.com/ |  |  |  |
| https://www.scjn.gob.mx/ |  |  |  |
| https://web.diputados.gob.mx/inicio |  |  |  |
| https://www.gob.mx/ |  |  |  |

1. https://search.google.com/test/mobile-friendly
2. https://ui.dev/amiresponsive

## Conceptos básicos de CSS3

Introducción

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para definir el aspecto y el diseño de páginas web escritas en lenguajes de marcado como HTML o XML. Su función principal es separar la estructura y contenido del documento del diseño y la presentación visual del mismo.

CSS es un lenguaje de programación muy poderoso que puede utilizarse para crear diseños web personalizados. Es fácil de aprender y usar, y es una parte esencial del desarrollo web.

Algunos ejemplos de lo que se puede hacer con CSS:

* Cambiar el tamaño y la fuente de la letra
* Cambiar el color de los elementos de la página
* Agregar fondos y bordes a los elementos de la página
* Crear animaciones y transiciones
* Crear diseños de página personalizados

Además de mejorar la consistencia y el aspecto visual, CSS también permite crear diseños responsivos y adaptables, lo que garantiza una experiencia de usuario óptima en diversos dispositivos, como computadoras de escritorio, tabletas y teléfonos móviles.

## Agregar estilos a un archivo HTML (Ejercicio)

1. Crear un archivo 06.css.html
2. Agrega el siguiente código HTML *(Si copia y pega, primero quité la numeración de Word).*

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <title>Ejemplo de Página Web</title>  </head>  <body>      <header>          <h1>Bienvenido a mi Página Web</h1>          <nav>              <ul>                  <li><a href="#inicio">Inicio</a></li>                  <li><a href="#servicios">Servicios</a></li>                  <li><a href="#contacto">Contacto</a></li>              </ul>          </nav>      </header>      <main>          <section id="inicio">              <h2>Sección de Inicio</h2>              <p>Bienvenido a mi página web. Esta es la sección de inicio. </p>          </section>          <section id="servicios">              <h2>Servicios</h2>              <ul>                  <li>Servicio 1</li>                  <li>Servicio 2</li>                  <li>Servicio 3</li>              </ul>          </section>      </main>      <footer>          <p>Contacto: info@misitio.com</p>      </footer>  </body>  </html> |

Incrustando estilos en el archivo HTML

1. Modifica el código y agrega la siguiente sección:

|  |
| --- |
| <title>Ejemplo de Página Web</title>  **<style>**          body {              font-family: Arial, sans-serif;              margin: 0;              padding: 0;              background-color: #f0f0f0;          }          header {              background-color: #333;              color: #fff;              padding: 10px;              text-align: center;          }          nav ul {              list-style: none;              margin: 0;              padding: 0;          }          nav li {              display: inline-block;              margin-right: 20px;          }          nav a {              color: #fff;              text-decoration: none;          }          main {              padding: 20px;          }          section {              margin-bottom: 30px;          }          footer {              background-color: #333;              color: #fff;              padding: 10px;              text-align: center;          }  **</style>** |

Vinculando estilos en archivos de hoja de estilos

1. Agrega el archivo 06.css.css con el siguiente contenido:

|  |
| --- |
| body {      font-family: Arial, sans-serif;      margin: 0;      padding: 0;      background-color: #f0f0f0;  }  header {      background-color: #333;      color: #fff;      padding: 10px;      text-align: center;  }  nav ul {      list-style: none;      margin: 0;      padding: 0;  }  nav li {      display: inline-block;      margin-right: 20px;  }  nav a {      color: #fff;      text-decoration: none;  }  main {      padding: 20px;  }  section {      margin-bottom: 30px;  }  footer {      background-color: #333;      color: #fff;      padding: 10px;      text-align: center;  } |

1. Modifica el archivo 06.css.html y cambia por la siguiente línea

|  |
| --- |
| <title>Ejemplo de Página Web</title>  <link rel="stylesheet" type="text/css" href="06.css.css" /> |

Estilos en línea a elementos HTML

1. Modifica el archivo 06.css.html y cambia por la siguiente línea

|  |
| --- |
| <header>      <h1 style="color:red">Bienvenido a mi Página Web</h1> |

**Importar la librería de estilos Bootstrap**

1. Modifica el archivo 06.css.html y cambia por la siguiente línea

|  |
| --- |
| <title>Ejemplo de Página Web</title>  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous"> |

1. Analiza el comportamiento.

## Selectores

Los selectores se utilizan para aplicar estilos a los elementos de un documento HTML. En CSS, un selector es una cadena de texto que identifica un elemento o un grupo de elementos en un documento HTML. En otras palabras, un selector CSS es el mecanismo que utilizamos para identificar y determinar qué elementos en el documento HTML serán afectados por las reglas de estilo que definimos.

Hay diferentes tipos de selectores CSS, cada uno de los cuales se utiliza para seleccionar diferentes tipos de elementos. Algunos de los tipos de selectores CSS más comunes son:

* **Selector de tipo o etiqueta:** Un selector de etiqueta selecciona todos los elementos de un tipo específico, independientemente de su ID o clase. Por ejemplo, el selector <div> seleccionará todos los elementos div en un documento HTML, independientemente de su ID o clase.
* **Selector universal:** El selector universal se indica con un asterisco (\*) y selecciona todos los elementos del documento (o del elemento padre si está encadenado con otro elemento y un operador de combinación descendente, por ejemplo).
* **Selector de clase:** Un selector de clase selecciona todos los elementos que tienen una clase específica. **Una clase** es un nombre que se asigna a uno o más elementos HTML. Por ejemplo, el **selector.myClass** seleccionará todos los elementos que tienen la clase **myClass**.
* **Selector de ID:** Un selector de ID selecciona un elemento específico por su ID. Un ID es un nombre único que se asigna a un elemento HTML. Por ejemplo, el selector #myDiv seleccionará el elemento div con el ID myDiv.

Los selectores CSS se pueden utilizar para aplicar estilos a los elementos de un documento HTML de muchas maneras diferentes. Por ejemplo, se pueden utilizar para cambiar el color, el tamaño y la fuente de los elementos, o para agregar fondos, bordes y animaciones a los elementos.

#### Ejercicio Selector de etiqueta.

Modifica el archivo 06.css.css con el siguiente contenido al final del archivo:

|  |
| --- |
| h2 {      /\* Estilos aplicados a todos los elementos <h2> (Encabezados) \*/      color: #ff0263;  } |

#### Ejercicio universal

Modifica el archivo 06.css.css con el siguiente contenido al principio del archivo:

|  |
| --- |
| \* {      color: green;  } |

#### Ejercicio de selector de clase

En un documento HTML, los selectores de clase buscan un elemento basado en el contenido de su atributo class. El atributo class está definido como una lista de elementos separados por espacio, y uno de esos elementos debe coincidir exactamente con el nombre de clase dado en el selector.

Modifica el archivo 06.css.css con el siguiente contenido al final del archivo (**selector. myClass)**:

|  |
| --- |
| p.myClass {      background-color: DodgerBlue;  } |

Modifica el archivo 06.css.html

|  |
| --- |
| <footer>          <p class="myClass">Contacto: info@misitio.com</p>      </footer> |

#### Ejercicio de selector de ID

En un documento HTML, los selectores de ID de CSS buscan un elemento basado en el contenido del atributo ID. **El atributo ID** del elemento seleccionado debe coincidir exactamente con el valor dado en el selector.

Modifica el archivo 06.css.css con el siguiente contenido al final del archivo:

|  |
| --- |
| #menu\_superior {      background-color: skyblue;  } |

Modifica el archivo 06.css.html

|  |
| --- |
| <header>     <h1 style="color:red">Bienvenido a mi Página Web</h1>     <nav id="menu\_superior"> |

#### Ejercicio selector hijo

El combinador > separa a dos selectores y busca solo a los elementos que coindicen con el segundo selector y que son hijos directos del primero.

**selector1 > selector2 { style properties }**

Modifica el archivo 06.css.css con el siguiente contenido al final del archivo:

|  |
| --- |
| #servicios > ul > li {      background-color: rgb(227, 241, 100);  } |

## Pseudoselectores

En CSS, los selectores que comienzan con dos puntos dobles (::) se conocen como "pseudo-elementos". Estos selectores permiten acceder y aplicar estilos a partes específicas de un elemento HTML que normalmente no se pueden seleccionar utilizando solo clases o identificadores. Los pseudo-elementos permiten una mayor especificidad en la aplicación de estilos a partes particulares de un elemento.

**Sintaxis:**

|  |
| --- |
| selector::pseudo-elemento { propiedad: valor; } |

* **::before:** Este pseudo-elemento se usa para insertar contenido antes del contenido real de un elemento. Es comúnmente utilizado para agregar contenido decorativo o elementos de diseño antes del contenido principal.

**Ejemplo:**

p::before {

content: "⭐ ";

}

* **::after: Similar a ::before,** este pseudo-elemento se usa para insertar contenido después del contenido real de un elemento.

**Ejemplo:**

p::after {

content: " 🌟";

}

* **::first-line:** Se utiliza para aplicar estilos a la primera línea de texto dentro de un elemento. Esto es útil para cambiar el estilo del primer párrafo o encabezado de un artículo, por ejemplo.

**Ejemplo:**

p::first-line {

font-weight: bold;

}

* **::first-letter:** Se utiliza para aplicar estilos a la primera letra de un bloque de texto.

**Ejemplo:**

p::first-letter {

font-size: 2em;

}

* **::selection:** Permite aplicar estilos al texto seleccionado por el usuario.

**Ejemplo:**

::selection {

background-color: yellow;

color: black;

}

* **:hover:**  aplicará un estilo cuando el usuario haga hover sobre el elemento especificado por el selector.

**Ejemplo:**

div:hover {

background-color: #F89B4D;

}

* **::placeholder:** Se utiliza para aplicar estilos al texto de marcador de posición (placeholder) dentro de un campo de entrada.

**Ejemplo:**

input::placeholder {

color: gray;

}

* **::before y ::after en contenido generado:** Estos pseudo-elementos también se pueden utilizar para generar contenido en el documento, no solo para estilizarlo. Por ejemplo, se pueden usar para agregar íconos o números a elementos de lista.

**Ejemplo:**

li::before {

content: "➜ ";

}

Los pseudo-elementos[[3]](#footnote-3) permiten una mayor flexibilidad en la creación de diseños y estilos detallados en tu sitio web. Es importante tener en cuenta que no todos los elementos admiten todos los pseudo-elementos, por lo que debes revisar la compatibilidad en diferentes navegadores si estás utilizando una amplia variedad de pseudo-elementos en tu diseño.

## Pseudo-classes

Una pseudoclase CSS es una palabra clave que se añade a los selectores y que especifica un **estado** especial del elemento seleccionado.

**Sintaxis:**

|  |
| --- |
| selector:pseudoclase { propiedad: valor; } |

Al igual que las clases, se pueden concatenar la cantidad de pseudoclases que se deseen en un selector.

Índice de las pseudo-clases estándar

:active

:checked

:default

:dir()

:disabled

:empty

:enabled

:first

:first-child

:first-of-type

:fullscreen

:focus

:hover

:indeterminate

:in-range

:invalid

:lang()

:last-child

:last-of-type

:left

:link

:not()

:nth-child()

:nth-last-child()

:nth-last-of-type()

:nth-of-type()

:only-child

:only-of-type

:optional

:out-of-range

:read-only

:read-write

:required

## :right

## :root

## :scope (en-US)

## :target

## :valid

## :visited

## Estructura de una regla CSS

La estructura completa es llamada regla predeterminada (pero a menudo «regla» para abreviar).

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* **Selector.** El elemento HTML en el que comienza la regla. Esta selecciona el(los) elemento(s) a dar estilo (en este caso, los elementos <p> ). Para dar estilo a un elemento diferente, solo cambia el selector.

Se utiliza el símbolo @ para definir reglas o directivas especiales que van más allá de la definición de estilos básicos para elementos HTML. Estas reglas especiales a menudo se usan para aplicar estilos avanzados, animaciones, transformaciones y otras características específicas. Algunas de las propiedades CSS que llevan el símbolo "@" son: @media:, @font-face:, @keyframes:, @import:, @supports:

* **Declaración.** Una sola regla como color: red; especifica a cuál de las propiedades del elemento quieres dar estilo.
* **Propiedades.** Maneras en las cuales puedes dar estilo a un elemento HTML. (En este caso, color es una propiedad del elemento <p> ). En CSS, seleccionas qué propiedad quieres afectar en tu regla.
  + **Propiedades abreviadas.** Permiten establecer los valores de varias otras propiedades CSS simultáneamente. Ej. background es una propiedad abreviada que puede definir los valores de background-color, background-image, background-repeat y background-position.
* **Valor de la propiedad.** A la derecha de la propiedad, después de los dos puntos (:), tienes el valor de la propiedad, para elegir una de las muchas posibles apariencias para una propiedad determinada (hay muchos valores para color además de red).

Sintaxis:

* Cada una de las reglas (aparte del selector) deben estar encapsuladas entre llaves ({}).
* Dentro de cada declaración, debes usar los dos puntos (:) para separar la propiedad de su valor.
* Dentro de cada regla, debes usar el punto y coma (;) para separar una declaración de la siguiente.

**Fuentes, colores, fondos y textos**

Estilos de texto

Las propiedades CSS que se usan para aplicar estilo al texto pueden clasificarse generalmente en dos categorías:

* **Estilos del tipo de letra:** Propiedades que afectan al texto (qué tipo de letra se usa, su tamaño, si es negrita, itálica, etc.).
* **Estilos de disposición del texto:** Propiedades que afectan al espaciado y otras características relativas a la disposición del texto, lo que permite la elección de, por ejemplo, el espacio entre líneas y letras, y el modo como el texto se alinea dentro de la caja contenedora.

La propiedad font-family para definir un tipo de letra diferente para nuestro texto. hay un conjunto de tipos de letra que están disponibles en todos los sistemas en general, y que en consecuencia son llamados tipos de letra seguros para la web, o web safe fonts: Arial, sans-serif; Courier New, monospace; Georgia, serif; Times New Roman, serif; Trebuchet MS, sans-serif; Verdana, sans-serif.

Ejemplo:

<div style="**color: red; font-size: 20px; font-family: sans-serif;**">

Este es un texto con estilo CSS en línea.

</div>

Colores

La propiedad color establece el color del contenido de los elementos seleccionados (que normalmente es texto, pero también puede incluir un par cosas más, como un subrayado o una línea superpuesta al texto con la propiedad text-decoration ).

El tipo de datos CSS <color> denota un color en el espacio de colores sRGB. Un color puede ser descrito de cualquiera de las siguiente maneras:

* Usando una palabra clave, son identificadores en los que no hay distinción entre mayúsculas y minúsculas y que sirven para representar un color concreto, por ejemplo red, blue, brown, lightseagreen.
* Usando el sistema RGB cubic-coordinate (usando #-hexadecimal o las notaciones funcionales rgb() y rgba() )
* Usando el sistema HSL cylindrical-coordinate (usando las notaciones funcionales hsl() y hsla()).

Ejemplos:

**/\* Cambiar el color del texto a rojo \*/**

**color: red;**

**/\* Usar un color hexadecimal \*/**

**color: #336699;**

**/\* Usar un nombre de color \*/**

**color: darkgreen;**

/\* Cambiar el color de fondo a amarillo \*/

background-color: yellow;

/\* Usar un color hexadecimal para el fondo \*/

background-color: #f0f0f0;

/\* Cambiar el color de los enlaces no visitados \*/

a:link { color: blue; }

/\* Cambiar el color de los enlaces visitados \*/

a:visited { color: purple; }

/\* Cambiar el color de los enlaces al pasar el cursor \*/

a:hover { color: orange; }

/\* Cambiar el color de los enlaces cuando se les da clic \*/

a:active { color: red; }

::placeholder {

color: lightgray;

}

Referencia

<a name="ancla-1"> . . . <>

<a href="#ancla-1">Capitulo 1</a>

border: <ancho> <estilo> <color>;

border: 1px solid red;

Fondos

La propiedad background es un atajo para definir los valores individuales del fondo en una única regla CSS. Se puede usar background para definir los valores de una o de todas las propiedades siguientes: background-attachment, color, image, position, repeat.

Links de apoyo:

* <https://www.css-gradient.com/>

Ejemplos:

h1 { background-color: red; }

h1 { background-image: url(fondo.jpg); }

h1 { background-repeat: repeat; }

h1 { background-position: top left; }

h1 { background-size: cover; }

/\* Crear un fondo con gradiente lineal \*/

background: linear-gradient(to right, #ff0000, #00ff00);

/\* Crear un fondo con gradiente radial \*/

background: radial-gradient(circle, #ff9900, #ff0000);

#### Ejercicio. Estilos del tipo de letra

El tamaño del tipo de letra se define con la propiedad font-size, se puede definir en al menos las siguientes unidades:

* **Unidades px (píxeles):** El número de píxeles de altura que deseas que tenga el texto. Esta es una unidad absoluta, da como resultado el mismo tamaño de letra en la página en casi cualquier situación.
* **Unidades em:** 1em equivale al tamaño de tipo de letra que se haya establecido en el elemento padre del elemento activo. Puedes tener un sitio web completo dimensionado con unidades em, lo que facilita su mantenimiento.
* **Unidades rem:** Funcionan igual que las unidades em, excepto que 1rem equivale al tamaño del tipo de letra establecido en el elemento raíz del documento (es decir, en **<html>**), no en el elemento padre.
* Modifica el archivo 06.css.css con el siguiente contenido en la sección correspondiente:

|  |
| --- |
| html {    font-size: 20px;  }  h1 {    font-size: 2rem;    }  h2 {    font-size: 1em;     }  p{    font-size: .5rem;     } |

El CSS proporciona cuatro propiedades comunes para alterar el efecto visual / énfasis del texto:

* **font-style:** Se usa para activar y desactivar el texto en cursiva. Los valores posibles son normal, italic, oblique.
* **font-weight:** Establece el grueso o peso del texto. Los valores posibles son normal, bold, lighter, bolder, 100–900.
* **text-transform:** Permite practicar algunas transformaciones sobre el tipo de letra. Los valores posibles: none, uppercase, lowercase, capitalize, full-width.

## Modelo de cajas y posicionamiento de elementos

El modelo de cajas de CSS es una forma de describir cómo se muestran los elementos HTML en una página web. Cada elemento HTML se representa como una caja, y el modelo de cajas describe el tamaño, la posición y la apariencia de estas cajas.

Al hacer una caja de tipo bloque en CSS tenemos cuatro partes:

* **Contenido (content)**: El contenido es el texto, las imágenes o cualquier otro contenido que se encuentra dentro de una caja.
* **Relleno (padding)**: El relleno es el espacio que se encuentra entre el contenido de una caja y sus bordes.
* **Margen (margin)**: El margen es el espacio que rodea a una caja y la separa de otras cajas.
* **Bordes (border)**: Los bordes son las líneas que rodean una caja.

Imagen que contiene Tabla

Descripción generada automáticamente

El modelo de cajas se puede utilizar para controlar el aspecto de los elementos HTML de muchas maneras diferentes. Por ejemplo, se puede utilizar para cambiar el tamaño de las cajas, para moverlas a diferentes ubicaciones en la página, o para agregar bordes o rellenos a las cajas.

#### Ejercicio. Modelo de cajas

* Crea el documento 08.cajas.html con el siguiente contenido

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Ejemplo Modelo de Cajas con colores</title>      <link rel="stylesheet" href="08.cajas.css">  </head>  <body>      <div class="caja">          <p>Contenido</p>      </div>  </body>  </html> |

* Crea el documento 08.cajas.css con el siguiente contenido

|  |
| --- |
| body {      margin: 0;      padding: 20px;      font-family: Arial, sans-serif;  }  .caja {      width: 200px;      height: 100px;      padding: 20px;      border: 5px solid #333;      margin: 40px;      background-color: #f0f0f0;  }  .caja p {      background-color: #ffc;  }  .caja::before {      content: "Contenido";      display: block;      background-color: #ccf;      border: 5px solid #00f;      padding: 20px;      margin: 20px;  } |

Overflow

La propiedad overflow se puede usar para controlar cómo se muestra el contenido de un elemento cuando se desborda del contenedor.

Se puede aplicar a elementos en línea y a elementos en bloque. Cuando se aplica a un elemento en línea, solo afecta al contenido del elemento. Cuando se aplica a un elemento en bloque, afecta al contenido del elemento y al espacio que ocupa el elemento en el contenedor.

La propiedad CSS overflow tiene los siguientes valores:

* **visible** - El contenido del elemento se muestra en su totalidad, incluso si se desborda del contenedor. Este es el valor predeterminado.
* **hidden** - El contenido del elemento que se desborda del contenedor se oculta.
* **scroll** - Se agregan barras de desplazamiento al contenedor para que el usuario pueda ver el contenido que se desborda.
* **auto** - Si el contenido del elemento se desborda del contenedor, se agregarán barras de desplazamiento automáticamente.
* **initial** - El valor de la propiedad se establece en su valor inicial.
* **inherit** - El valor de la propiedad se hereda del elemento padre.

Posicionamiento

El posicionamiento de CSS es una forma de controlar la posición de los elementos HTML en una página web. El posicionamiento se puede utilizar para crear diseños web más complejos y personalizados.

Hay cuatro tipos de posicionamiento de CSS:

* **Posicionamiento absoluto:** El posicionamiento absoluto coloca un elemento en una posición fija en la página, independientemente de la posición de los otros elementos.
* **Posicionamiento relativo:** El posicionamiento relativo coloca un elemento en una posición relativa a su posición original.
* **Posicionamiento flotante:** El posicionamiento flotante coloca un elemento en una capa flotante que se desplaza por encima de los demás elementos.
* **Posicionamiento normal:** El posicionamiento normal es el posicionamiento predeterminado de los elementos HTML. Los elementos se posicionan en su orden de aparición en el documento HTML.

Posicionamiento

Gráfico, Diagrama

Descripción generada automáticamente

Fuente: https://www.csssolid.com/css-position.html

#### Ejercicio. Modelo de cajas

Utilizar las herramientas de desarrollo del navegador para ver el modelo de cajas.

#### Ejercicio. Posicionamiento

* Crea el documento 09.posicionamiento.html con el siguiente contenido

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Ejemplo Posicionamiento</title>      <link rel="stylesheet" href="09.posicionamiento.css">  </head>  <body>      <div class="contenedor">          <div class="caja-normal">Caja 1 (Posicionamiento normal)</div>          <div class="caja-flotante">Caja 2 (Posicionamiento flotante)</div>          <div class="caja-absoluta">Caja 3 (Posicionamiento absoluto)</div>          <div class="caja-relativa">Caja 4 (Posicionamiento relativo)</div>      </div>  </body>  </html> |

* Crea el documento 09.posicionamiento.css con el siguiente contenido

|  |
| --- |
| body {      margin: 0;      padding: 20px;      font-family: Arial, sans-serif;  }  .contenedor {      position: relative;      width: 400px;      background-color: #f0f0f0;      border: 2px solid #333;      padding: 10px;  }  .caja-normal {      width: 300px;      height: 100px;      background-color: #ccffcc;      border: 1px solid #0f0;      padding: 5px;      margin-bottom: 20px;  }  .caja-flotante {      width: 100px;      height: 100px;      background-color: #ffcccc;      border: 1px solid #f00;      padding: 5px;      float: left;      margin-right: 10px;  }  .caja-absoluta {      position: absolute;      top: 50px;      left: 150px;      width: 200px;      height: 80px;      background-color: #ffffcc;      border: 1px solid #ff0;      padding: 5px;  }  .caja-relativa {      position: relative;      width: 150px;      height: 150px;      background-color: #ccccff;      border: 1px solid #00f;      padding: 5px;  } |

Utilizar las herramientas de desarrollo del navegador para ver el modelo de posicionamiento.

Interfaz de usuario gráfica, Diagrama, Aplicación

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

## Maquetación

Los elementos semánticos de HTML5 son etiquetas específicas que se utilizan para estructurar y proporcionar significado a las partes del contenido de una página web de una manera más clara y semántica

Estos elementos semánticos hacen que el código HTML sea más expresivo, facilitan la comprensión y la navegación del contenido, y mejoran la accesibilidad y la optimización para motores de búsqueda. Es recomendable utilizarlos adecuadamente para estructurar y organizar el contenido de manera más significativa y eficiente.

Algunos ejemplos de elementos semánticos de HTML5 son:

* <header>: Representa el encabezado de una sección o página, generalmente contiene logotipos, títulos, menús de navegación, etc.
* <nav>: Define una sección de navegación, como un menú de navegación.
* <main>: Indica el contenido principal del documento. Debe contener el contenido central de la página.
* <section>: Define una sección temática de contenido. Se utiliza para agrupar contenido relacionado.
* <article>: Representa un contenido independiente y completo, como una publicación de blog, un artículo de noticias, una publicación en un foro, etc.
* <aside>: Contiene contenido relacionado o complementario que se encuentra alrededor del contenido principal.
* <footer>: Representa el pie de página de una sección o página. Generalmente contiene información de contacto, enlaces, derechos de autor, etc.
* <figure>: Utilizado para agrupar contenido relacionado con imágenes, diagramas, gráficos, etc.
* <figcaption>: Proporciona una leyenda o descripción para una figura (contenido dentro de <figure>).
* <time>: Utilizado para representar fechas y horas en un formato legible por máquinas y humanos.
* <details>: Crea un widget que muestra u oculta detalles adicionales cuando se hace clic.
* <summary>: Proporciona un título o resumen para el contenido de <details>.
* <mark>: Destaca o resalta el texto para llamar la atención del lector.
* <blockquote>: Utilizado para citas largas o textos citados de otras fuentes.
* <cite>: Indica la referencia a una obra o autor citado en un texto.
* <abbr>: Define una abreviatura o acrónimo.
* <address>: Utilizado para proporcionar información de contacto o dirección.

Interfaz de usuario gráfica

Descripción generada automáticamente

## Maquetación con porcentajes

La maquetación CSS con porcentajes se refiere a utilizar unidades de porcentaje (%) en lugar de unidades fijas (como píxeles) para definir tamaños y posiciones de elementos HTML en una página web. Cuando se utilizan porcentajes en CSS, los valores se calculan en relación con el tamaño de su elemento padre o contenedor en lugar de utilizar valores absolutos.

Al utilizar porcentajes en la maquetación CSS, se logra un diseño más flexible y adaptable a diferentes tamaños de pantalla y dispositivos. Esto es especialmente útil para crear diseños responsivos, que se ajustan automáticamente a diferentes resoluciones y dimensiones de pantalla.

Para utilizar la maquetación CSS con porcentajes, es necesario definir los anchos y alturas de los elementos de la interfaz de usuario en términos de porcentajes del ancho y alto de la ventana del navegador. Por ejemplo, para definir un elemento para que ocupe el 50% del ancho de la ventana del navegador, se puede utilizar la siguiente propiedad CSS:

width: 50%;

También es posible utilizar la maquetación CSS con porcentajes para definir los espaciados entre los elementos de la interfaz de usuario. Por ejemplo, para definir un espaciado de 10 píxeles entre dos elementos, se puede utilizar la siguiente propiedad CSS:

padding: 10px;

#### Ejercicio. Maquetación con porcentajes

* Crea el documento 10.porcentajes.html con el siguiente contenido

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Maquetación con Porcentajes</title>      <link rel="stylesheet" href="10.porcentajes.css">  </head>  <body>      <div class="contenedor">          <div class="caja1">Caja 1</div>          <div class="caja2">Caja 2</div>      </div>  </body>  </html> |

* Crea el documento 10.porcentajes.css con el siguiente contenido

|  |
| --- |
| body {      margin: 0;      padding: 20px;      font-family: Arial, sans-serif;  }  .contenedor {      width: 80%; /\* El contenedor ocupa el 80% del ancho de la ventana del navegador \*/      margin: 0 auto; /\* Centra el contenedor horizontalmente \*/      background-color: #f0f0f0;      padding: 10px;  }  .caja1 {      width: 50%; /\* La caja ocupa el 50% del ancho de su contenedor \*/      padding: 10px;      background-color: #ffcccc;      margin-bottom: 10px;  }  .caja2 {      width: 30%; /\* La caja ocupa el 30% del ancho de su contenedor \*/      padding: 10px;      background-color: #ccffcc;  } |

Utilizar las herramientas de desarrollo del navegador para analizar y modificar el ejemplo.

En este ejemplo, el contenedor ocupa el 80% del ancho de la ventana del navegador (width: 80%;). Luego, las cajas dentro del contenedor se dimensionan utilizando porcentajes en relación con el ancho de su contenedor. La caja1 ocupa el 50% del ancho del contenedor, y la caja2 ocupa el 30% del ancho del contenedor. Esto asegura que ambas cajas se ajusten automáticamente en función del tamaño del contenedor y de la ventana del navegador.

#### Ejercicio. Maquetación con porcentajes (HTML Semántico).

* Crea el documento 11.porcentajes2.html con el siguiente contenido

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Ejemplo Maquetación con Porcentajes</title>      <link rel="stylesheet" href="11.porcentajes2.css">  </head>  <body>      <header>          <h1>Encabezado de la Página</h1>      </header>      <nav>          <ul>              <li><a href="#">Inicio</a></li>              <li><a href="#">Acerca de</a></li>              <li><a href="#">Servicios</a></li>              <li><a href="#">Contacto</a></li>          </ul>      </nav>      <main>          <section class="contenido">              <h2>Contenido Principal</h2>              <p>Este es el contenido principal de la página.</p>          </section>          <aside class="sidebar">              <h2>Barra Lateral</h2>              <p>Esta es una barra lateral con contenido adicional.</p>          </aside>      </main>      <footer>          <p>Derechos de Autor © 2023 | Mi Sitio Web</p>      </footer>  </body>  </html> |

* Crea el documento 11.porcentajes2.css con el siguiente contenido

|  |
| --- |
| body {      margin: 0;      padding: 0;      font-family: Arial, sans-serif;  }  header {      background-color: #333;      color: #fff;      text-align: center;      padding: 10px;  }  nav {      background-color: #444;      color: #fff;      text-align: center;      padding: 10px;  }  nav ul {      list-style: none;      margin: 0;      padding: 0;  }  nav li {      display: inline-block;      margin-right: 20px;  }  nav a {      color: #fff;      text-decoration: none;  }  main {      width: 80%;      margin: 0 auto;      padding: 20px;      display: flex;      flex-wrap: wrap;  }  section.contenido {      width: 70%;      padding: 10px;      background-color: #f0f0f0;      margin-right: 20px;  }  aside.sidebar {      width: 30%;      padding: 10px;      background-color: #f8f8f8;  }  footer {      background-color: #333;      color: #fff;      text-align: center;      padding: 10px;  } |

El uso de porcentajes en la maquetación CSS permite que los elementos se escalen proporcionalmente y se adapten a diferentes pantallas, lo que contribuye a la creación de un diseño más responsivo y amigable para múltiples dispositivos.

## Media queries

La maquetación CSS con media queries es una técnica utilizada para aplicar estilos diferentes a una página web en función de las características del dispositivo o del medio en el que se está visualizando. Con las media queries, los desarrolladores web pueden crear diseños y estilos responsivos que se ajusten automáticamente a diferentes tamaños de pantalla, dispositivos y condiciones de visualización.

Las media queries permiten establecer reglas de CSS específicas para diferentes condiciones, como el ancho y alto de la pantalla, la orientación (horizontal o vertical), la densidad de píxeles, la resolución, el tipo de dispositivo (pantalla, impresora, etc.) y otras características.

Para utilizar la maquetación CSS con media queries, es necesario definir las reglas de estilo en bloques de código llamados media queries. Cada media query se define por una condición, que es una combinación de valores de ancho de pantalla, orientación de pantalla y densidad de píxeles.

**Sintaxis de una media query:**

|  |
| --- |
| @media media-type and (media-feature) {  /\* Reglas CSS que se aplicarán cuando se cumplan las condiciones de la media query \*/  } |

Donde:

**media-type:** Representa el tipo de dispositivo o medio, como screen (pantalla), print (impresora), speech (dispositivos de síntesis de voz), entre otros.

**media-feature:** Especifica la condición para la cual se aplicarán las reglas de CSS. Por ejemplo, max-width, min-width, orientation, resolution, etc.

Por ejemplo, el siguiente media query se aplica a las páginas web que tienen un ancho de pantalla de 1024 píxeles o más:

|  |
| --- |
| @media (min-width: 1024px) {      .container {        width: 50%;      }    } |

También es posible utilizar media queries para definir diferentes estilos de presentación para diferentes dispositivos. Por ejemplo, el siguiente media query se aplica a las páginas web que se muestran en una tableta:

|  |
| --- |
| @media (min-width: 768px) and (max-width: 1024px) {      .container {        width: 75%;      }    } |

#### Ejercicio. Maquetación con media queries

* Crea el documento 12.media.html con el siguiente contenido

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Ejemplo Maquetación con Media Queries</title>      <link rel="stylesheet" href="12.media.css">  </head>  <body>      <header>          <h1>Encabezado de la Página</h1>      </header>      <nav>          <ul>              <li><a href="#">Inicio</a></li>              <li><a href="#">Acerca de</a></li>              <li><a href="#">Servicios</a></li>              <li><a href="#">Contacto</a></li>          </ul>      </nav>      <main>          <section class="contenido">              <h2>Contenido Principal</h2>              <p>Este es el contenido principal de la página.</p>          </section>          <aside class="sidebar">              <h2>Barra Lateral</h2>              <p>Esta es una barra lateral con contenido adicional.</p>          </aside>      </main>      <footer>          <p>Derechos de Autor © 2023 | Mi Sitio Web</p>      </footer>  </body>  </html> |

* Crea el documento 12.media.css con el siguiente contenido

|  |
| --- |
| body {      margin: 0;      padding: 0;      font-family: Arial, sans-serif;  }  header {      background-color: #333;      color: #fff;      text-align: center;      padding: 10px;  }  nav {      background-color: #444;      color: #fff;      text-align: center;      padding: 10px;  }  nav ul {      list-style: none;      margin: 0;      padding: 0;  }  nav li {      display: inline-block;      margin-right: 20px;  }  nav a {      color: #fff;      text-decoration: none;  }  main {      width: 80%;      margin: 0 auto;      padding: 20px;      display: flex;      flex-wrap: wrap;  }  section.contenido {      width: 70%;      padding: 10px;      background-color: #f0f0f0;      margin-right: 20px;  }  aside.sidebar {      width: 30%;      padding: 10px;      background-color: #f8f8f8;  }  footer {      background-color: #333;      color: #fff;      text-align: center;      padding: 10px;  }  /\* Media Query para pantallas con un ancho máximo de 600px \*/  @media screen and (max-width: 600px) {      main {          flex-direction: column;      }      section.contenido {          width: 100%;          margin-right: 0;          margin-bottom: 20px;      }      aside.sidebar {          width: 100%;      }  } |

## Viewport

El viewport en CSS responsivo se refiere a la ventana de visualización o área visible de un dispositivo, como la pantalla de un navegador web o el espacio disponible para mostrar contenido en un dispositivo móvil o tableta. En el contexto de diseño web responsivo, el viewport es una parte esencial para crear diseños que se adapten a diferentes tamaños de pantalla y dispositivos.

Cuando se carga una página web en un navegador, el tamaño del viewport puede variar según el dispositivo en el que se está visualizando. En los dispositivos móviles, por ejemplo, el viewport suele ser más pequeño que en las computadoras de escritorio.

Para asegurar que una página web se muestre correctamente en diferentes dispositivos, es fundamental definir cómo se debe ajustar el contenido dentro del viewport. Para lograr esto, se utiliza una etiqueta meta llamada "viewport" en el encabezado HTML del documento.

Sintaxis de la etiqueta meta viewport:

<meta name="viewport" content="propiedad-valor">

Las propiedades más comunes utilizadas en la etiqueta "viewport" son:

* width: Define el ancho del viewport. Por ejemplo, width=device-width establece el ancho del viewport como el ancho del dispositivo.
* initial-scale: Especifica el nivel de zoom inicial cuando se carga la página. Por ejemplo, initial-scale=1.0 representa el 100% de zoom.
* minimum-scale: Define el nivel mínimo de zoom permitido.
* maximum-scale: Define el nivel máximo de zoom permitido.
* user-scalable: Indica si el usuario puede realizar zoom en la página. Por ejemplo, user-scalable=no deshabilita el zoom.

#### Ejercicio. Viewport

* Crea el documento 13.viewport.html con el siguiente contenido

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Página Web Responsiva</title>  </head>  <body>      <!-- Contenido de la página -->  </body>  </html> |

## Proyecto del alumno

1. BootstrapForma, Círculo

   Descripción generada automáticamente

## Introducción

Bootstrap es un popular framework front-end de código abierto desarrollado por Twitter. Fue creado con el objetivo de facilitar y agilizar el proceso de diseño y desarrollo de sitios web y aplicaciones web responsivas. Bootstrap ofrece una colección de herramientas, estilos predefinidos y componentes reutilizables que permiten a los desarrolladores crear interfaces modernas y atractivas de manera eficiente.

Con Bootstrap, no es necesario comenzar desde cero. El framework proporciona una estructura de base sólida y un sistema de cuadrícula flexible que facilita la organización y alineación de los elementos en la página. Además, incluye una amplia variedad de componentes, como botones, formularios, navegación, tarjetas, carruseles y más, que pueden ser fácilmente personalizados y adaptados a las necesidades de cada proyecto.

La característica más destacada de Bootstrap es su enfoque en la responsividad. Los diseños creados con Bootstrap se ajustan automáticamente a diferentes tamaños de pantalla, lo que garantiza una experiencia de usuario consistente en dispositivos móviles, tabletas y computadoras de escritorio.

Bootstrap proporciona una amplia gama de componentes de diseño, que incluyen:

* Botones
* Cajas de texto
* Encabezados
* Pies de página
* Menús
* Formularios
* Tablas
* Gráficos

Bootstrap también proporciona una amplia gama de estilos, que se pueden utilizar para personalizar el aspecto de tu sitio web.

## Proyecto del alumno

1. jQuery o JavascriptForma, Círculo

   Descripción generada automáticamente

## Introducción

jQuery es una biblioteca de JavaScript que facilita el trabajo con el DOM (Document Object Model) y los eventos de JavaScript. Es una biblioteca muy popular y utilizada por millones de desarrolladores web en todo el mundo.

jQuery se basa en JavaScript, pero proporciona una API más sencilla y fácil de usar. Esto lo hace ideal para los desarrolladores web que no están familiarizados con JavaScript o que quieren acelerar el desarrollo de sus sitios web.

jQuery proporciona una amplia gama de funciones, que incluyen:

* Selección de elementos del DOM
* Adición, eliminación y modificación de elementos del DOM
* Manejo de eventos
* Animación
* AJAX

Para usar jQuery en WordPress, primero debes asegurarte de que la versión de jQuery que deseas utilizar esté disponible en tu instalación de WordPress. A partir de la versión 5.7, WordPress incluye jQuery por defecto.

Primero, necesitas agregar la biblioteca jQuery a tu sitio web de WordPress. Puedes hacerlo agregando el siguiente código a tu archivo functions.php:

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

Una vez que hayas agregado jQuery a tu sitio web, puedes empezar a usarlo para agregar funcionalidad a tu página web. Por ejemplo, puedes usar jQuery para agregar animaciones a tus elementos HTML, o para agregar eventos a tus elementos HTML.

|  |
| --- |
| <script>      $(document).ready(function() {        $("#myElement").animate({          width: "100px",          huich: "100px"        }, 1000);      });      </script> |

El código anterior agregará una animación al elemento HTML con el ID "myElement". La animación aumentará el ancho y el alto del elemento a 100 píxeles en un segundo.

Ejemplo de cómo usar jQuery para agregar un evento a un elemento HTML:

|  |
| --- |
| <script>      $(document).ready(function() {        $("#myButton").on("click", function() {          alert("Hello World!");        });      });      </script> |

El código anterior agregará un evento al elemento HTML con el ID "myButton". El evento se disparará cuando el usuario haga clic en el botón. El evento mostrará un cuadro de diálogo con el mensaje "¡Hola Mundo!".

## Proyecto del alumno

1. Conectar a APIsForma, Círculo

   Descripción generada automáticamente

## Introducción

En WordPress, puedes utilizar Ajax para realizar solicitudes asincrónicas al servidor sin tener que recargar completamente la página. Esto te permite actualizar partes específicas de la página de forma dinámica y mejorar la experiencia del usuario. Aquí te doy un ejemplo simple de cómo usar Ajax en WordPress:

Agrega el siguiente código en el archivo functions.php de tu tema o en tu plugin personalizado para registrar y encolar el script Ajax:

|  |
| --- |
| function my\_enqueue\_scripts() {      wp\_enqueue\_script( 'ajax-script', get\_template\_directory\_uri() . '/js/ajax-script.js', array( 'jquery' ), '1.0', true );      // Pasar el valor de la URL de Ajax al script      wp\_localize\_script( 'ajax-script', 'ajax\_object', array( 'ajax\_url' => admin\_url( 'admin-ajax.php' ) ) );  }  add\_action( 'wp\_enqueue\_scripts', 'my\_enqueue\_scripts' );  // Procesar la solicitud Ajax  function my\_ajax\_function() {      // Aquí puedes realizar tus operaciones con Ajax      // Por ejemplo, obtener datos de la base de datos o procesar formularios.      // Retorna la respuesta en formato JSON      $response = array(          'message' => '¡Esta es una respuesta de Ajax desde WordPress!',      );      wp\_send\_json( $response );  }  add\_action( 'wp\_ajax\_my\_ajax\_function', 'my\_ajax\_function' );  add\_action( 'wp\_ajax\_nopriv\_my\_ajax\_function', 'my\_ajax\_function' ); |

Crea un archivo llamado ajax-script.js dentro de la carpeta "js" de tu tema o plugin con el siguiente código:

|  |
| --- |
| jQuery(document).ready(function($) {      // Ejecutar el código de Ajax cuando se haga clic en un botón      $('#my-ajax-button').on('click', function() {          // Realizar la solicitud Ajax          $.ajax({              url: ajax\_object.ajax\_url,              type: 'post',              data: {                  action: 'my\_ajax\_function' // Nombre del gancho de acción para procesar la solicitud Ajax              },              success: function(response) {                  // Mostrar la respuesta en un elemento HTML                  $('#ajax-response').html(response.message);              },              error: function(error) {                  console.log(error);              }          });      });  }); |

En cualquier parte de tu contenido, puedes agregar un botón con un identificador #my-ajax-button y un elemento con el identificador #ajax-response para mostrar la respuesta del servidor. Por ejemplo:

|  |
| --- |
| <button id="my-ajax-button">Haz clic para solicitar Ajax</button>  <div id="ajax-response"></div> |

En este ejemplo, al hacer clic en el botón con el identificador #my-ajax-button, se ejecutará una solicitud Ajax al servidor a través de la función my\_ajax\_function. La respuesta del servidor se mostrará en el elemento con el identificador #ajax-response.

## Proyecto del alumno

1. Experiencia de usuarioForma, Círculo

   Descripción generada automáticamente

## Principio de Proximidad

Se refiere a la tendencia natural del ojo humano a agrupar elementos que están cerca entre sí. Esto se debe a que nuestro cerebro interpreta los elementos cercanos como unidades relacionadas, independientemente de sus formas o colores. Cuando varios elementos se encuentran próximos, se convierten en una unidad visual única.

Este principio juega un papel crucial en la organización y la estructura visual de una composición. Cuando los elementos están agrupados por proximidad, se facilita la comprensión y el reconocimiento de patrones, permitiendo a los espectadores interpretar la información de manera más rápida y eficiente.

Al comprender cómo funciona el ojo humano, los diseñadores pueden utilizar el principio de proximidad para crear diseños que sean más atractivos, fáciles de navegar y visualmente atractivos.

Algunas formas de aplicar el principio de proximidad incluyen:

* Agrupar elementos relacionados, como títulos y texto, para formar bloques cohesivos.
* Utilizar el espacio en blanco para crear separaciones entre elementos.
* Jugar con diferentes tamaños y tipos de letra para crear una sensación de jerarquía.
* Utilizar el color para crear contraste y llamar la atención.
* Colocar imágenes y sus respectivos títulos cercanos uno del otro para indicar su asociación.
* Organizar elementos en una cuadrícula, donde los elementos del mismo grupo se alinean en filas o columnas.
* Usar márgenes y espacios en blanco para separar diferentes secciones y evitar la sensación de aglomeración.

Ejemplos:

En el sitio web de Uber, si bien la navegación y el titular los percibimos como un grupo, no sucede lo mismo con los dos bloques de debajo. Están tan separados entre sí que los vemos como entidades distintas. Y lo mismo sucede con los botones, que deberían ir con cada bloque[[4]](#footnote-4):

Interfaz de usuario gráfica

Descripción generada automáticamente

**Ejercicio:**

Examina el siguiente sitio web y contesta las siguientes preguntas:

* ¿Es fácil encontrar la información?
* ¿Tus ojos se sienten cómodos siguiendo la información?
* ¿Sientes orden y armonía en el sitio?

## Principio de la Alineación

Es un principio de organización visual que establece que los elementos de un diseño deben estar alineados de manera consistente. Esto puede crear una sensación de orden y cohesión en un diseño, y también puede facilitar la lectura de un diseño.

Cuando los elementos están bien alineados, se crea una sensación de armonía y equilibrio visual, lo que facilita la comprensión y la navegación para el espectador:

* Mejora la legibilidad y la comprensión: Una alineación adecuada facilita la lectura del contenido y la interpretación de la información.
* Genera cohesión: Los elementos alineados crean una sensación de unidad y coherencia en el diseño.
* Crea una apariencia profesional: El diseño alineado da una impresión de profesionalismo y pulcritud.
* Ayuda a establecer jerarquías visuales: La alineación puede utilizarse para resaltar elementos clave y establecer una jerarquía en la información presentada.

Algunas formas de aplicar el principio de alineación incluyen:

* Alineación con la cuadrícula: Se basa en una estructura de cuadrícula invisible que divide el espacio en filas y columnas. Los elementos se alinean con estas líneas para lograr una distribución coherente.
* Alineación con márgenes: Los elementos se alinean con los márgenes del diseño, como el borde izquierdo o derecho de una página o el margen superior o inferior.
* Alineación central: Los elementos se centran en relación con un punto central, lo que crea un equilibrio simétrico.
* Alineación jerárquica: En algunos casos, ciertos elementos pueden estar alineados con respecto a otros elementos principales para establecer una jerarquía visual clara.

Aquí hay algunos consejos para usar el principio de alineación en diseño gráfico:

* Usa la alineación para crear una sensación de orden y cohesión en un diseño.
* Usa la alineación para facilitar la lectura de un diseño.
* Usa la alineación para crear una sensación de jerarquía en un diseño.
* Experimenta con diferentes tipos de alineación para crear diferentes efectos.

Ejemplos:

|  |  |
| --- | --- |
|  |  |
| Fuente: Pinterest.com | |

En diseños donde el número de elementos es elevado, la alineación cobra aún más importancia sobre todo si hay mucho texto. En el siguiente ejemplo, la perfecta alineación de los elementos, junto con un buen empleo del espacio, facilita de manera notoria la legibilidad de la información que se nos está mostrando[[5]](#footnote-5) :

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

## Principio de la Repetición

Es un principio de organización visual que establece que los elementos de un diseño deben repetirse de manera consistente. Se refiere a la utilización intencionada y repetida de elementos visuales similares a lo largo de una composición. Estos elementos pueden incluir formas, colores, tipografías, patrones u otros elementos gráficos. Esto puede crear una sensación de orden, cohesión y ritmo en un diseño, y también puede facilitar la lectura de un diseño.

La repetición se utiliza para crear consistencia, cohesión y unidad en un diseño. Cuando elementos similares se repiten a lo largo del diseño, se establece una relación visual entre ellos, lo que a su vez ayuda a transmitir un mensaje claro y atractivo. Algunos de los beneficios clave del principio de repetición incluyen:

* Unidad visual: La repetición ayuda a unificar una composición, creando una sensación de cohesión y organización. Los elementos repetidos actúan como elementos de enlace que conectan diferentes partes del diseño.
* Fácil comprensión: Al repetir ciertos elementos, se facilita la comprensión y la navegación a través del diseño. La audiencia identificará rápidamente los patrones y entenderá la estructura visual de la composición.
* Fortalecimiento de la marca: En el contexto del diseño de marcas, la repetición de elementos visuales como logotipos, colores o tipografías específicas ayuda a establecer una identidad de marca consistente y reconocible.
* Enfatizar jerarquías: La repetición puede utilizarse para destacar ciertos elementos y establecer jerarquías visuales. Elementos que se repiten con mayor frecuencia tienden a recibir más atención y se perciben como más importantes.
* Atractivo estético: La repetición bien ejecutada puede resultar en un diseño atractivo y armonioso, ya que los elementos similares juntos crean una sensación de ritmo y equilibrio.

Algunas formas de aplicar el principio de repetición incluyen a repetición de elementos que pueden ser:

* Colores
* Formas
* Tamaños
* Tipografía
* Patrones
* Imágenes

Ejemplos:

Repetir el mismo elemento una y otra vez crea un patrón que es altamente satisfactorio para el ojo. Además de esto, la repetición ayuda a crear consistencia, un factor crucial cuando se trata de imponer la unidad de tu composición, o tu identidad como marca[[6]](#footnote-6).

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

## Principio de Contraste

Es un principio de organización visual que establece que los elementos de un diseño deben ser diferentes entre sí. Se refiere a la diferencia notable y deliberada entre elementos visuales dentro de una composición. Esto puede crear una sensación de interés, énfasis y jerarquía en un diseño, y también puede facilitar la lectura de un diseño. Estos elementos contrastantes pueden incluir colores, formas, tamaños, tipografías, texturas y cualquier otro atributo gráfico.

El contraste es una herramienta poderosa que ayuda a captar la atención del espectador y a destacar ciertos elementos importantes en un diseño. Al crear diferencias visuales significativas entre elementos, se establece una jerarquía y se guía la mirada del espectador hacia lo que realmente importa en la composición.

Algunas formas de aplicar el principio de contraste incluyen:

* Contraste de colores: Utilizar colores opuestos en el círculo cromático para resaltar elementos clave o para generar un fuerte impacto visual.
* Contraste de tamaño: Colocar elementos significativos mucho más grandes o más pequeños que otros para enfatizar su importancia o crear una sensación de escala.
* Contraste de formas: Utilizar formas diferentes y distintivas para destacar elementos específicos dentro del diseño.
* Contraste de tipografías: Utilizar tipografías de estilos diferentes (por ejemplo, una tipografía sans-serif junto a una tipografía serif) para diferenciar títulos, subtítulos y texto normal.
* Contraste de texturas: Combinar texturas o patrones distintos para resaltar ciertas áreas o elementos del diseño.

Usar el principio de contraste, los diseñadores gráficos pueden crear diseños que sean visualmente atractivos, fáciles de leer y fáciles de entender. Aquí hay algunos consejos para usar el principio de contraste en diseño gráfico:

* Usa el contraste para crear una sensación de interés en un diseño.
* Usa el contraste para crear una sensación de énfasis en un diseño.
* Usa el contraste para crear una sensación de jerarquía en un diseño.
* Experimenta con diferentes tipos de contraste para crear diferentes efectos.

Ejemplos:

Imagen que contiene Logotipo

Descripción generada automáticamente

Trip Advisor hace un buen uso del color para hacer que los resultados de búsqueda resalten y sean vibrantes, de este modo habilita a los usuarios para una eficiente selección y clasificación de data más importante que se les presenta[[7]](#footnote-7).

Cuando estás buscando hoteles en cualquier locación, Trip Advisor solo busca las mejores opciones y las muestra, pero los resultados son más atractivos porque están presentados en una paleta de color en tríada presentes en los resultados, esto mejora significativamente la experiencia de usuario. Esta paleta de color está basada en usar una misma paleta de colores que naturalmente están del lado opuesto en el círculo cromático entre sí.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

## Cuadricula

En teoría del diseño, una cuadrícula es una estructura organizativa y geométrica que se utiliza para dividir el espacio de un diseño en columnas y filas equidistantes. Esta herramienta proporciona una guía visual que ayuda a los diseñadores a organizar y distribuir elementos de manera coherente y equilibrada en una composición gráfica.

Las cuadrículas pueden ayudar a los diseñadores a crear diseños que sean visualmente atractivos, fáciles de leer y fáciles de entender.

* Las cuadrículas pueden ayudar a los diseñadores a organizar elementos de diseño y crear un diseño equilibrado y simétrico.
* Las cuadrículas pueden ayudar a los diseñadores a guiar el ojo del espectador a través de un diseño.
* Las cuadrículas pueden ayudar a los diseñadores a crear diseños que sean visualmente atractivos y fáciles de leer.
* Las cuadrículas pueden ayudar a los diseñadores a crear diseños que sean fáciles de adaptar a diferentes tamaños y formatos.

Ejemplos

**Columnas, espaciado entre columnas y márgenes**

La cuadrícula de diseño adaptable de Material Design y Bootstrap se componen de tres elementos: columnas, medianeras y márgenes.

Gráfico

Descripción generada automáticamente

El número mágico de columas es 12.

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

Un dibujo de una persona

Descripción generada automáticamente con confianza media

## Jerarquía Visual

La jerarquía visual en diseño web se refiere a la organización y presentación visual de los elementos en una página web, con el objetivo de guiar la atención del usuario y comunicar la importancia relativa de cada elemento en la página. Es una herramienta clave para facilitar la comprensión y la navegación del contenido, ya que permite a los usuarios identificar rápidamente qué elementos son más relevantes y cómo están relacionados entre sí.

La jerarquía visual se logra mediante el uso estratégico de varios elementos de diseño, como tamaño, color, contraste, tipografía y posición en la página. Algunas técnicas comunes para establecer una jerarquía visual efectiva en diseño web son:

* Tamaño: Los elementos más importantes se suelen mostrar en un tamaño más grande que los elementos secundarios o menos relevantes.
* Contraste: El uso de colores contrastantes o de un esquema de color diferente para destacar ciertos elementos puede ayudar a establecer jerarquías visuales claras.
* Tipografía: Utilizar diferentes tamaños, pesos y estilos de tipografía para diferenciar títulos, subtítulos y texto normal.
* Espacio en blanco: El uso estratégico de espacios en blanco o "aire" entre elementos también puede ayudar a resaltar áreas clave y mejorar la legibilidad.
* Posición y alineación: Colocar elementos importantes en posiciones destacadas, como en la parte superior de la página o en el centro, y alinearlos para crear una estructura visual coherente.
* Imágenes y elementos visuales: Utilizar imágenes llamativas o elementos gráficos para atraer la atención del usuario hacia ciertas áreas.

Una jerarquía visual efectiva ayuda a los usuarios a:

* Encontrar la información que necesitan rápidamente
* Entender la estructura de la página web
* Seguir las instrucciones de la página web
* Navegar por la página web de manera fluida

Ejemplo

Una correcta jerarquía visual en el diseño web organiza los contenidos de la interfaz de manera más eficaz y también son más atractivos para los usuarios. Es un concepto clave que los diseñadores siempre prestan mucha atención para mejorar la experiencia de usuario al agregar elementos visuales para destacar sus sitios web.

La jerarquía visual en el diseño web se refiere a la disposición y presentación de los elementos de la interfaz en orden de importancia para que los usuarios puedan escanear fácil y rápidamente la información que necesitan, hacer clic y comprar un producto o servicio del sitio web sin problemas[[8]](#footnote-8).

En la jerarquía visual se aplican los principios de diseño de Proximidad, Contraste,

El tamaño, como una de las herramientas más poderosas para una mejor jerarquía visual, tiene un gran impacto en la visualización de los elementos de una página web. Cuanto más grande sea el elemento (como fotos, textos y formas), más atención captará.

Colores, Los colores brillantes siempre llaman la atención de los usuarios de forma más eficaz y rápida respecto a los colores apagados. Por lo tanto, usa colores brillantes para enfatizar los contenidos importantes de la web.

Imagen que contiene persona, mujer, firmar, sostener

Descripción generada automáticamente

Degradados, pueden ayudar a crear una jerarquía visual clara para un sitio web.

Patrones de escaneo, según el informe de Nielsen Norman Group, existen varios patrones de escaneo visual mientras las personas navegan por páginas web, como el patrón “F” o “Z”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Independientemente de los anterior, las personas usarán los patrones de escaneo “F” o “Z”, a menudo leen tanto el encabezado como la parte inferior de la página con atención para buscar la información principal de izquierda a derecha. Y al igual que con el contenido del cuerpo, a menudo escanearán la jerarquía (que consta de mosaicos web y subtítulos) y elegirán partes útiles solo para leer.

La jerarquía fallida

Una buena jerarquía visual no se trata de gráficos salvajes y locos o de los más recientes filtros de Photoshop, sino de organizar la información de manera que sea utilizable, accesible y lógica para el visitante diario del sitio*[[9]](#footnote-9)*.

Motor de un carro

Descripción generada automáticamente con confianza media

Espacio, especialmente el espacio en blanco (también llamado espacio negativo), entre los elementos de la interfaz de usuario del sitio web también puede ayudar a resaltar el contenido y crear una jerarquía visual de manera intuitiva.

Interfaz de usuario gráfica

Descripción generada automáticamente

## Combinación de colores

Tipografía

## Tipografía

1. WordpressForma, Círculo

   Descripción generada automáticamente

## Introducción

WordPress es una plataforma de gestión de contenido (CMS, por sus siglas en inglés) de código abierto, ampliamente utilizada para crear sitios web, blogs y aplicaciones web. Lanzado en 2003, WordPress ha evolucionado desde una simple plataforma de blogging hasta convertirse en un robusto y versátil CMS que alimenta una gran parte de la web actual.

Una de las principales razones de la popularidad de WordPress es su facilidad de uso. No se requiere conocimientos avanzados de programación para empezar a utilizarlo. Cuenta con una interfaz intuitiva y amigable, lo que permite a usuarios sin experiencia técnica gestionar y actualizar su contenido de manera sencilla.

WordPress se basa en un sistema de temas y complementos que extienden su funcionalidad. Los temas permiten personalizar la apariencia y el diseño del sitio web, mientras que los complementos (plugins) agregan características adicionales, como formularios de contacto, galerías de imágenes, optimización para motores de búsqueda (SEO), integración con redes sociales y mucho más.

Otra característica destacada de WordPress es su enfoque en la optimización para motores de búsqueda (SEO). El código limpio y la estructura de URLs amigables que ofrece WordPress contribuyen a mejorar la visibilidad de los sitios en los resultados de búsqueda.

Además, WordPress es una comunidad muy grande y activa. Cuenta con una gran cantidad de desarrolladores y diseñadores que crean temas y complementos, ofreciendo a los usuarios una amplia variedad de opciones para personalizar sus sitios web según sus necesidades.

## Proyecto del alumno – Landing page

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

## Instalación

|  |  |
| --- | --- |
| Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente | /wp-admin/ |
| Pantalla de Instalación | Configuración a la base de datos |

## Interfaz

|  |  |
| --- | --- |
|  |  |
| Escritorio (Panel de Control)  Zona privada | Frontend  Zona pública |

Escritorio (Panel de Control)

* Menú principal
  + Escritorio
  + **Entradas**
  + Medios
  + **Páginas**
  + Comentarios
  + **Apariencia**
  + Plugins
  + Usuaarios
  + Herramientas
  + Ajustes
  + Abrir y cerrar menú
* Escritorio
* Opciones de pantalla
* Nombre del sitio
* Vínculo a la sección pública del proyecto

## Ajustes de configuración

* **Ajustes generales**
  + Título del sitio
  + Descripción corta
  + Dirección de WordPress (URL)
  + Dirección del sitio (URL)
  + Dirección de correo electrónico de administración
  + Miembros
* Ajustes de escritura
* Ajustes de lectura
  + Tu página de inicio muestra
  + Número máximo de entradas a mostrar en el sitio
  + Número máximo de entradas a mostrar en el feed
  + Para cada entrada en el feed, incluir (<https://localhost/wp/index.php/feed/>)
  + Visibilidad en los motores de búsqueda
* Ajustes de comentarios
* Ajustes de medios
* Ajustes de los enlaces permanentes
* Privacidad

## Paginas y entradas en WordPress

Las páginas y las entradas son dos de los tipos de contenido más comunes en WordPress.

Las entradas son unidades de contenido cronológico, comúnmente utilizadas en blogs para compartir noticias, actualizaciones y opiniones, mientras que las páginas son para contenido estático y permanente, como información institucional o páginas de referencia.

**Páginas**

* Las páginas son contenido estático que no cambia con el tiempo.
* Las páginas se suelen utilizar para crear contenido como la página de inicio, la página de contacto y las páginas de productos o servicios.
* Las páginas se pueden insertar en el menú de navegación de tu sitio web.

**Entradas**

* Las entradas son contenido dinámico que se actualiza con el tiempo.
* Las entradas se suelen utilizar para crear blogs, noticias y otros tipos de contenido que se actualizan regularmente.
* Las entradas se pueden mostrar en la página de inicio de tu sitio web y en las páginas de archivo de entradas.
* Las entradas permiten asociar Categorías y Etiquetas, mientras que las páginas no.

## Partes de una página

* Cabecera
* Contenido (content)
* Barras laterales (sidebar’s)
  + Widgets
* Footer
  + Widgets, menú secundarios

## Menu

* Menú Principal

## Tipos de usuarios

* Administrador
* Editor
* Autor
* Contribuidor
* Suscriptor

## Edición de páginas (Configuración)

**Página**

* Visibilidad
  + Borrador
  + Pendiente de revisión
  + Publicada
* Programar publicación
* Plantilla de la página
* Url
* Autor
* Imagen destacada
* Comentarios
* Atributos de página
  + Página superior

**Bloque**

## Instalación de plantillas

* Apariencia
  + Temas
  + GeneratePress
* Opciones de personalización del tema
  + Posiciones de las barras de contenido
* Wingets

**Instalación de Elementor**

* Instalar elementor
* Instalar Essential Addons for Elementor

**Edición con Elementor**

**Ejercicio práctico – Wordpress - Elementor**

Ejercicio – Construir la página principal de un sitio web

https://iaenelperiodismo.diputados.gob.mx/

1. Reiniciar la instalación de wordpress

*Actualizaciones / Reinstalar*

*Configuración*

*-- Título del sitio: Beneficios de la IA para el Periodismo*

*Usuario: admio/123*

1. Instalar los archivos de medios
2. Instalar el tema

*Apariencia / Tema / Añadir Nuevo / Generate Press / Activar*

1. Crear las páginas, con la propiedad "Atributos de Página/ Disposición de la barra lateral / Sin barras laterales"

*Inicio*

*Registro*

*Ponentes*

*Galeria*

*Aviso de privacidad*

*------*

*Mandar a la papelera: Página de ejemplo, Política de privacidad — Página de Política de Privacidad*

1. Ajustes / Lectura

*Tu portada muestra:*

*-- Una página estática: Portada / Inicio*

*-- Página de entradas / Blog*

1. Instalar plugins

*Plugins / Añadir Nuevo / Elementor Page/\WebSite Builder Versión 3.15.2 / Instalar / Activar /*

*--- Conectar con una cuenta de correo para que poder usar elementor*

*Plugins / Añadir Nuevo / Essential Addons for Elementor Versión 5.8.6 |/ Instalar / Activar*

*Plugins / Añadir Nuevo / 3D FlipBook - Lite Edition Versión 1.15.2 / Instalar / Activar*

1. Actualizar página "Atributos de Página" Elementor ancho completo

*Página / Inicio / Elementor ancho completo*

1. 5.- Editar la primera página en modo elementor

**## Inicio**

Modificar / Página / Plantilla / Elementor Ancho Completo

--> Ver el sitio, para comprobar que no se ve ningun contenido

**Editar con Elementor**

Revisar el sitio de ejemplo:

**## Imagen Principal**

Sección de bloques /

Agregar Widget Elementor / Filtrar / Features / Seleccionar la que tiene una barra de imagen

---> Actualizar ver la página

Sustituir la imagen de fondo

--> Click en el contenedor

Altura mínima 800

Forma del divisor/Tipo, hacer pruebas con las opciones.

Arriba/ Tipo / Olas

Abajo/Tipo / Nubes

--> Cilck en propiedades de la columna

--> Click en propiedades de espaciador

**## Objetivo**

Sección de bloques /

Agregar Widget Elementor / Filtrar / About Company / Selecionar la que se parece más al ejemplo:

- Sustituir el texto

- OBJETIVO / Etiqueta ***HTML H2***

Editar Columna / Avanzado / Animación de Entrada 1.25 segs

- Remover el botón READ MORE

- Sustituir la imagen

**## PDF**

Salir del editor de Elementor

*- Seleccionar en el menú la opción 3D FlipBook / Añadir Nueva*

*- Source: PDF*

*- Select PDF*

*- Publicar*

*- Copiar el ShortCode*

*Opcionalmente. 3DFlipBook / Shortcode Generator / Seleccionar el Programa*

*- Copiar el ShorCode*

Regresar a editar el Página/Inicio en Elementor

*- Agregar un renglon*

*- En los Widgets, buscar shortcode*

*- Arrastrar al simbolo +*

*- Pegar el shorcode que trae en el portapapeles*

1. Personalizar el encabezado, fuentes y colores

Apariencia / Personalizar

Identidad del Sitio

- Titulo del Sitio / Ocultar título del sitio

- Elegir Logotipo

- Elegir Icono del Sitio

------------------------

Diseño de pantalla

- Cabecera / Actual

- Ancho del encabezado / Completo

- Alineación de la cabecera / Centrar

Diseño de pantañña

Navegación Principal / Sin cambios

Diseño de pantalla / Pie de página

Boton Volver arriba / Activado

-------------------------------

Colores

**Navegación:**

**#7A7A7A**

**#537463**

----------------------------

Tipografía:

Gestor de Fuentes: Roboto

Gestor de Tipografía

Elementos del menu principal / Familia de fuentes: Roboto / Grosor: Negrita / Tamaño 16Px

1. Editar con Elementor Ajustes del Sitio

*Registro*

*Ponentes*

*Galeria*

*Aviso de privacidad*

1. *El tema es gratuito y no permite cambiar el pie de página.*

*Solución*

*El propio tema proporciona la solución en* [*https://docs.generatepress.com/article/changing-the-copyright-message/*](https://docs.generatepress.com/article/changing-the-copyright-message/)

*1.- Instalar y activar el Plugin Code Snippets By Code Snippets Pro*

*2.- Crear un snipets:*

*Nombre: cambiar\_generate\_copyright*

*Code:*

|  |
| --- |
| *add\_filter( 'generate\_copyright','tu\_custom\_copyright' );*  *function tu\_custom\_copyright() {*  *?>*  *Cámara de Diputados © 2023 | LXV La legislatura de la paridad, la inclusión y la diversidad.*  *<?php*  *}* |

*Guardar -> Activar*

1. ***Cambiar los colores del pie de página***

*Personalización / Colores / Barra de Pie de Página*

*Fondo: var(--contrast)*

*Texto: var(--base-3)*

1. ***Personalizar el encabezado, para simular el encabezado fijo***

*Personalización / CSS Adicional*

|  |
| --- |
| *.site-header {*  *z-index:1;*  *position: fixed;*  *left: 0;*  *right: 0;*  *height: 92px;*  *padding: 0px;*  *padding-left: 20px;*  *color: midnightblue;*  *background-color: white;*  *border-color: red;*  *}* |

***Página REGISTRO***

1. *Página / Registro / Página / Plantilla / Elementor ancho completo*

*--> Ver el sitio, para comprobar que no se ve ningún contenido*

*Editar con Elementor*

*Revisar el sitio de ejemplo:*

1. Agregar bloque de imagen

Sección de bloques /

Agregar Widget Elementor / Filtrar / About / Seleccionar la que se parece al boceto

1. Instalar el Plugin Formulario Contact Form 7 o *WP Forms*

Contacto / Añadir Nuevo

1. *Abrir Navigator / Seleccionar el primer componente / Boton Derecho / Guardar como plantilla.*
2. *Editar página ponentes*

*Insertar desde la plantilla anterior*

*Sustituir los contenidos*

1. Revisar para mas detalle: https://developer.mozilla.org/es/docs/Learn/HTML [↑](#footnote-ref-1)
2. Fuente: https://developer.mozilla.org/es/docs/Learn/HTML/Introduction\_to\_HTML/Getting\_started [↑](#footnote-ref-2)
3. https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-elements [↑](#footnote-ref-3)
4. Crpeditos https://www.uifrommars.com/principios-gestalt-diseno-web/ [↑](#footnote-ref-4)
5. Fuente: https://harphymurx.com/principios-del-diseno-grafico/ [↑](#footnote-ref-5)
6. Fuente: https://es.wix.com/blog/2018/08/principios-del-diseno-aplicados-a-la-web/ [↑](#footnote-ref-6)
7. https://www.staffdigital.pe/blog/diseno-web-contraste-experiencia-usuario/ [↑](#footnote-ref-7)
8. Fuente: https://helensotillo.com/web/consejos-y-ejemplos-para-una-correcta-jerarquia-visual-en-el-diseno-web/ [↑](#footnote-ref-8)
9. https://webdesign.tutsplus.com/es/understanding-visual-hierarchy-in-web-design--webdesign-84a [↑](#footnote-ref-9)