



Especialidad Técnico en: INFORMÁTICA

Módulo: HTML (HyperText Markup Language)

HTML



Avenida Venustiano Carranza No 20, Cabecera municipal de Chimalhuacán, Estado de México

Teléfonos 58-52-33-01 (o) 15-51-66-37

Escuela de Capacitación de Chimalhuacán



Índice

1. Introducción a programación web	5
1.1 ¿Qué es programación?	5
1.2 Importancia de la programación web	6
1.3 ¿Qué es HTML?	7
1.4 Historia de HTML.....	7
1.5 ¿Cómo crear un archivo HTML?	8
1.6 Estructura básica de HTML.....	9
1.6.1 Elementos semánticos.....	9
1.7 Atributos en HTML	10
1.8 Normas y Reglas de HTML.....	11
2. Texto en HTML.....	12
2.1 Insertar texto en un documento HTML.....	12
Práctica 1: Mi primera página web.....	13
2.2 Párrafos y salto de línea	13
2.3 Alineamiento de texto.....	14
Práctica 2: Refranes	15
2.4 Textos prediseñados.....	15
Práctica 3: Texto predise.....	16
Práctica 4: Diferencias	16
2.5 Negrita, Cursiva y Subrayado.....	17
Práctica 5: Vocales.....	17
2.6 Títulos.....	18
Práctica 6: Escribe tu nombre.....	18
2.7 Fuente de texto	19
Práctica 7: Colores	19
Práctica 8: Poemas	20
3. Imágenes	20
3.1 Tamaño de la imagen	21
Practica 1: Inserta 10 imágenes.....	22
3.2 Alineación.....	22
Practica 2: Noticia.....	22
3.3 Imagen de fondo	22
4. Marquesinas.....	23
Práctica 1: Ejemplo de marquesina	23
6.1 Direcciones de marquesinas.....	23

Práctica 2: Diferentes marquesinas	23
6.2 Velocidades de marquesinas	24
Práctica 3: 10 imágenes.....	24
5. Listas	24
5.1 Listas Ordenadas	24
Práctica 1: Canciones.....	25
5.2 Listas No ordenadas	25
Práctica 2: Reglas del salón	25
5.3 Listas De definición.....	26
Práctica 3: Receta	26
6. Tablas.....	27
Práctica 1: Tabla 1.....	28
Práctica 2: Tabla 2.....	28
6.1 Combinar celdas.....	28
Práctica 3: Tabla 3.....	30
7. Enlaces.....	30
Práctica 1: 5 historias	30
Practica 2: Índice	30
8. Audio	31
Práctica 1: Canción con letra	31
Práctica 2: Rockola.....	32
9. Video	32
Práctica 1: Video con letra.....	33
Práctica 2: Película.....	33
10. Formularios	34
10.1 Form	35
10.1.1 Input <input />	35
10.1.2 Atributos comunes del Formulario.....	35
10.1.2.1 El atributo “type”	35
10.1.2.2 El atributo name	35
10.1.2.3 El atributo value	35
10.2 Cajas de texto	36
10.3 Botones	36
Práctica 1: Formulario 1	37
10.4 Botones radio	37
10.5 Casillas de verificación.....	38

Práctica 2: Formulario 2	38
10.6 Área de texto	39
Práctica 3: Formulario 3	39
10.7 Menú de selección	40
10.8 Fecha y hora	41
10.9 Números	41
10.10 Validación de datos para formularios.....	42
10.10.1 Expresiones regulares para <input> (pattern).....	43
Práctica 4: Formulario 4	44
11. Práctica final	44
12. Evaluación.....	45

1. Introducción a programación web

Este curso introductorio a la programación web está diseñado para proporcionarle una comprensión sólida de los conceptos esenciales para la creación de sitios web interactivos y atractivos. Exploraremos los lenguajes de programación fundamentales, las tecnologías clave y las mejores prácticas para desarrollar sitios web efectivos desde cero. Además, exploraremos cómo estos conocimientos se relacionan con el mundo del marketing en línea, permitiéndole crear sitios web que no solo sean funcionales, sino también efectivos en la promoción de productos y servicios.

1.1 ¿Qué es programación?

La programación, en el contexto de la informática y la tecnología, se refiere al proceso de crear conjuntos de instrucciones o código que le indican a una computadora cómo realizar una tarea específica. Estas instrucciones están escritas en lenguajes de programación, que son conjuntos de reglas y sintaxis comprensibles tanto para los humanos como para las máquinas.

La programación es esencial para desarrollar software, aplicaciones y sistemas informáticos de todo tipo. A través de la programación, los desarrolladores pueden crear desde simples scripts hasta complejas aplicaciones de software, plataformas en línea y sistemas de control industrial. Los programas informáticos son creados para automatizar tareas, procesar datos, realizar cálculos, interactuar con usuarios y mucho más.

El proceso de programación generalmente implica los siguientes pasos:

Planificación: Definir el problema que se resolverá o la tarea que se automatizará. Esto implica comprender los requisitos y objetivos del programa.

Diseño: Crear un diseño conceptual de cómo funcionará el programa, identificando las estructuras de datos necesarias y los algoritmos que se utilizarán.

Codificación: Traducir el diseño en código de programación válido. Aquí es donde se escriben las instrucciones detalladas que la computadora ejecutará.

Prueba: Probar el programa para asegurarse de que funcione correctamente y cumpla con los requisitos. Esto puede implicar identificar y corregir errores o problemas (bugs).

Depuración: Corregir cualquier error o fallo encontrado durante las pruebas.

Optimización: Realizar mejoras en el código para que el programa funcione de manera más eficiente y rápida, si es necesario.

Documentación: Crear documentación clara y comprensible para otros desarrolladores o usuarios que puedan necesitar comprender o mantener el programa en el futuro.

1.2 Importancia de la programación web

La programación web desempeña un papel fundamental en la actualidad debido a la creciente importancia de Internet y la presencia en línea en todos los aspectos de la vida. Aquí hay algunas razones clave que destacan la importancia de la programación web:

Presencia en Línea: La programación web permite crear y mantener presencia en línea a través de sitios web y aplicaciones web. En una era en la que la mayoría de las personas buscan información y realizan transacciones en línea, tener una presencia en la web se ha convertido en una necesidad para individuos, empresas, organizaciones y gobiernos.

Accesibilidad Global: Los sitios web y las aplicaciones web son accesibles desde cualquier lugar del mundo con acceso a Internet. Esto brinda la posibilidad de llegar a audiencias globales y ampliar el alcance de las comunicaciones y los negocios.

Interacción y Comunicación: La programación web permite la interacción en línea, ya sea a través de formularios de contacto, sistemas de comentarios, chat en vivo o redes sociales integradas. Esto fomenta la comunicación bidireccional entre los usuarios y los propietarios del sitio.

Comercio Electrónico: La programación web es esencial para el comercio electrónico, permitiendo la creación de tiendas en línea, carritos de compra, procesamiento de pagos y gestión de inventario. Esto ha revolucionado la forma en que las personas compran y venden productos y servicios.

Marketing Digital: Las estrategias de marketing digital, como la optimización para motores de búsqueda, la publicidad en línea y las campañas de redes sociales, dependen de la programación web para implementarse efectivamente.

Experiencia del Usuario: Los sitios web modernos buscan proporcionar una experiencia de usuario fluida y atractiva. La programación web permite crear interfaces interactivas y amigables que mantienen a los usuarios comprometidos.

Análisis y Seguimiento: La programación web permite la integración de herramientas de análisis, que brindan información detallada sobre el comportamiento de los usuarios en el sitio. Esto es crucial para tomar decisiones informadas y mejorar el rendimiento del sitio.

Automatización: La programación web puede automatizar tareas repetitivas y procesos, lo que ahorra tiempo y reduce errores humanos. Por ejemplo, los formularios en línea pueden automatizar la recopilación de datos y el envío de respuestas.

Innovación Tecnológica: La programación web es una industria en constante evolución, lo que lleva a la creación de nuevas tecnologías y enfoques para mejorar la experiencia en línea. Esto incluye el desarrollo de aplicaciones web progresivas, sitios responsivos, chatbots en sitios web, entre otros.

Colaboración y Educación: La programación web también se utiliza en plataformas de colaboración en línea y en sistemas de educación a distancia. Permite la creación de entornos virtuales donde las personas pueden interactuar, aprender y trabajar juntas,

independientemente de su ubicación física.

1.3 ¿Qué es HTML?

HTML (HyperText Markup Language o Lenguaje de Marcado de Hipertexto), suele simplificarse a que es el lenguaje con el que se elaboran páginas web/páginas de internet. En principio se aclarará que HTML no es un lenguaje de programación, aunque de forma coloquial muchas veces se oigan referencias a HTML como si lo fuera. HTML es un lenguaje de etiquetas. Estas etiquetas (tag) HTML comunican al navegador cuál es la información para mostrar por pantalla, además del formato de dicha información, es un lenguaje de maquetación web o lenguaje de etiquetas destinado a crear estructuras de documentos HTML.

1.4 Historia de HTML

El origen de HTML se remonta a 1980, cuando el físico Tim Berners-Lee, propuso un nuevo sistema de "hipertexto" para compartir documentos.

En el ámbito de la informática, el "hipertexto" permitía que los usuarios accedieran a la información relacionada con los documentos electrónicos que estaban visualizando.

Después de unir sus fuerzas con el ingeniero de sistemas Robert Cailliau, presentaron la propuesta ganadora llamada WorldWideWeb (W3).

El primer documento formal con la descripción de HTML se publicó en 1991 bajo el nombre HTML Tags (Etiquetas HTML) y todavía hoy puede ser consultado online a modo de reliquia informática.

En 1995, el organismo IETF organiza un grupo de trabajo de HTML y consigue publicar, el 22 de septiembre de ese mismo año, el estándar HTML 2.0. A pesar de su nombre, HTML 2.0 es el primer estándar oficial de HTML.

En el año 1997 HTML 4.0 se introdujo con mejoras significativas en la estructura y la presentación. Se incorporaron marcos (frames), hojas de estilo en línea (inline styles) y mejoras en la accesibilidad.

Para el año 2000 se estableció XHTML (Extensible HyperText Markup Language) como una evolución de HTML, siguiendo pautas más estrictas de XML. XHTML buscaba una mayor coherencia y compatibilidad con otras tecnologías.

En 2014 HTML5 se completó oficialmente como un estándar por el W3C (World Wide Web Consortium). Esta versión marcó un gran avance en la funcionalidad web y abrió puertas a la creación de aplicaciones web más ricas y complejas, introdujo nuevas etiquetas semánticas, elementos multimedia (como <audio> y <video>), capacidades de arrastrar y soltar, y una mayor integración de JavaScript.

1.5 ¿Cómo crear un archivo HTML?

Existen diferentes maneras de crear un documento con extensión HTML, ya sean con procesadores de texto, editores de texto o compiladores (pequeños programas informático, que se encargan de traducir o compilar el código fuente de cualquier aplicación que se esté desarrollando), cada uno con sus características específicas, pero al final el diseño de la página depende de la persona que la diseña y no de la aplicación que se use.

En el curso utilizaremos un editor de texto; Bloc de notas, ya que requiere más habilidad y mejor manejo de conocimiento, pues este está centrado en el manejo de las etiquetas en vez de aprender las herramientas de la aplicación.

Para crear su primer archivo en HTML, realice lo siguiente:

1.- Abrir Bloc de notas de Windows, seleccione el botón Inicio, escriba Bloc de notas y luego seleccione Bloc de notas en los resultados.

2.- Una vez abierto el programa, escriba las siguientes líneas:

```
<html>
<head>
<title>Prueba_1</title>
</head>
<body>
<p>¡Hola mundo!</p>
</body>
</html>
```

3.- Para guardar el documento, se dirigirá a la barra de menús, seleccione Archivo, seleccione Guardar como..., elija el lugar donde guardará el documento, coloque un nombre para el documento y al final del nombre coloque .html con el fin de cambiar la extensión del archivo (por ejemplo index.html), una vez hecho solo de clic en Guardar.



4.- Para poder visualizar la página web, abra el Explorador de archivos de Windows, seleccione el botón Inicio, escriba Explorador de archivos y luego seleccione Explorador de archivos en los resultados, para después dirigirse a la ubicación donde guardo su archivo con extensión .html que acaba de crear. Una vez que lo haya ubicado, de doble clic sobre este, normalmente se abrirá en su Navegador web predeterminado.

1.6 Estructura básica de HTML

Las etiquetas que están continuación se escriben en el orden que se muestra en un bloc de notas

Etiquetas	Función
<html>	Esta etiqueta comienza el archivo y especifica que es un documento HTML.
<head>	Es la cabecera nos dará información sobre el documento y todo lo referente al title, styles vinculados, scripts, etc.
<title>	Entre esta etiqueta pondremos el título de nuestra página web que aparecerá en la pestaña de nuestro navegador.
</title>	Etiqueta de cierre de title.
</head>	Etiqueta de cierre de head.
<body>	Es el cuerpo de nuestra página, aquí irá toda la información que nuestro usuario podrá ver, es la parte más importante de nuestro documento y dentro de body se pueden usar la mayoría de las etiquetas con las que cuenta HTML.
</body>	Etiqueta de cierre de body.
</html>	Etiqueta de cierre de html, usualmente fin del documento.

Recuerda: Apréndete estas etiquetas muy bien, pues la mayoría de los documentos HTML están conformados de éstas y en ese orden.

1.6.1 Elementos semánticos

Los elementos semánticos en HTML son etiquetas que proporcionan un significado adicional y estructura semántica al contenido de una página web. Estos elementos no solo definen cómo se ve el contenido, sino también qué significado tiene en el contexto del documento. Utilizar elementos semánticos adecuados no solo mejora la legibilidad y accesibilidad del código, sino que también ayuda a los motores de búsqueda a entender mejor el contenido de la página. Aquí hay una descripción de algunos de los elementos semánticos más comunes:

<header>: Define la cabecera de una sección o de la página completa. Por lo general, contiene elementos como títulos, logotipos y elementos de navegación.

<nav>: Indica que el contenido es una sección de navegación. Suele contener enlaces a otras partes del sitio web.

<main>: Define el contenido principal de una página. Debería ser único y contener el contenido central de la página.

<article>: Representa un contenido independiente y autocontenido, como una publicación de blog, un artículo de noticias o un comentario.

<section>: Define una sección de contenido relacionado temáticamente. Puede contener encabezados, párrafos y otros elementos.

<aside>: Representa contenido que es tangencial al contenido principal y puede contener información como barras laterales, anuncios o contenido relacionado.

<footer>: Define el pie de página de una sección o de la página completa. Suele contener información de contacto, enlaces a páginas relacionadas y derechos de autor.

<figure>: Utilizado para encapsular contenido multimedia, como imágenes, gráficos o videos, junto con su descripción usando la etiqueta **<figcaption>**.

<figcaption>: Proporciona una descripción o leyenda para el contenido dentro del elemento **<figure>**.

<time>: Representa información de fecha y hora. Puede contener fechas, horas o intervalos de tiempo, y puede ser útil para motores de búsqueda y usuarios.

<mark>: Resalta parte del texto para enfatizar su relevancia o para identificar partes específicas de un texto.

<blockquote>: Indica una cita larga de otra fuente. Puede contener tanto texto como elementos HTML.

<cite>: Usado para citar el título de una obra creativa, como un libro o una película, dentro de un elemento **<blockquote>** o **<q>**.

<q>: Indica una cita corta dentro del texto. Puede ser utilizado junto con el elemento **<cite>** para proporcionar la fuente de la cita.

<address>: Utilizado para mostrar información de contacto, como direcciones de correo electrónico o números de teléfono, en un contexto de contenido.

Estos elementos semánticos ayudan a estructurar y organizar el contenido de una manera que sea significativa para los humanos y las máquinas, mejorando la accesibilidad, el SEO y la comprensión general de la página web.

1.7 Atributos en HTML

Los atributos en HTML son características que tienen las etiquetas y que configuran el valor de éstas o su aspecto visual. Cualquier etiqueta HTML puede contener uno o más atributos, separados por espacios, que permiten especificar la etiqueta. El código suele estar conformado de esta manera:

```
<body bgcolor = "red">
```

En la etiqueta **<body>**, estamos especificando el color de fondo con el atributo **"bgcolor"** (**bgcolor = color de fondo**), que en este caso será **"red"** (rojo).

Este solo es un ejemplo, pero la mayoría de las etiquetas de HTML tienen atributos para modificar con la excepción de la etiqueta **<html>**.

1.8 Normas y Reglas de HTML

HTML tiene una sintaxis muy sencilla y fácil de comprender, el contenido está estructurado en elementos, los cuales son declarados mediante el uso de etiquetas que proporcionan información adicional a través de atributos.

Si seguimos una correcta codificación no tendremos ningún problema y obtendremos todos los beneficios del lenguaje.

Hay que recordar que hay etiquetas que pueden modificar todo el texto, para tener mayor control de nuestro diseño y evitar modificar alguna parte que no debemos, hay que recordar siempre cerrar las etiquetas que abramos, tomemos la siguiente línea para explicar esto:

La etiqueta `<a>` que estamos utilizando es de anclaje, link o también llamado hipervínculo, sirve para direccionar al usuario hacia alguna otra dirección o página web y esta etiqueta modifica al texto, imagen o elemento que le procede en este caso la palabra "YouTube", entonces para evitar convertir toda tu página en un gigantesco hipervínculo es necesario cerrar la etiqueta `<a>` con una etiqueta de cierre y esto no es más que utilizar una barra diagonal "/" acompañado de la etiqueta inicial `<a>` para conseguir `` como en el ejemplo.

```
<a href="https://www.youtube.com"> YouTube </a>
```

Los atributos de las etiquetas van al interior de la etiqueta principal, el atributo "href" se encarga de establecer una propiedad que en este caso es el hipervínculo y esta es dictada con el símbolo "=" el cual nos indica que el valor del atributo será el siguiente, y, por último, pero no menos importante, el valor del atributo siempre deberá estar entre comillas dobles como en el ejemplo, "https://www.youtube.com" que nos guía hacia la dirección que especificamos.

Dijimos al principio que HTML es un lenguaje de etiquetas y esto implica que el programa hará las cosas secuencialmente, esto significa que habrá que mantener siempre el orden, pues los elementos deberán estar correctamente anidados, es decir que los elementos incluidos dentro de otros se cierren correctamente antes de cerrar el elemento padre; podemos tener líneas como esta:

```
<b><i> Hola Richard</i></b>
```

Ya que hablamos de la importancia de las etiquetas de cierre, en este caso témenos dos etiquetas la primera `` para indicar un texto en negrita y la segunda `<i>` que sirve para mostrar el texto en cursiva, ambas con sus respectivas etiquetas de cierre. Para tener una buena jerarquía de etiquetas será necesario cerrar en orden las etiquetas, cerrando primero la etiqueta `<i>` y al final ``.

Para facilitar las cosas hay que tener en cuenta que las etiquetas se cierran en un orden inverso al que usamos, cerrando desde la última etiqueta que abres hasta la primera.

Por último, recuerda que las etiquetas siempre serán escritas en minúsculas, aunque hay navegadores que aceptan minúsculas y mayúsculas el lenguaje está

diseñado a base de etiquetas completamente en minúscula para mejorar la legibilidad y coherencia del código.



`Recuerda`
`<h3>Escribir en</h3>`
`Minúsculas`



`No`
`<H3>Escribas en</H3>`
`Mayúsculas`

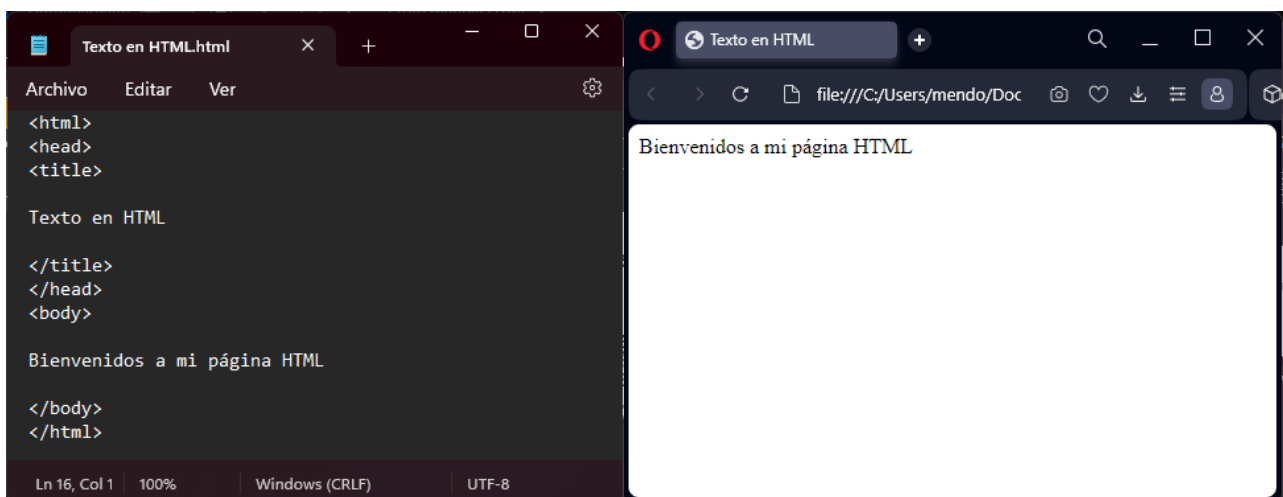
2. Texto en HTML

El texto desempeña un papel fundamental en HTML y en el diseño de sitios web en general. Aunque los elementos visuales y multimedia son esenciales, el contenido textual sigue siendo la columna vertebral de cualquier página web, pues es la forma principal en que se comunica la información, se optimiza para motores de búsqueda, se mejora la accesibilidad y se crea una experiencia de usuario efectiva en la web. Es un componente clave para lograr los objetivos de comunicación, marketing y diseño de un sitio web.

2.1 Insertar texto en un documento HTML

Introducir texto en HTML es de las cosas más básicas y sencillas, aun así, muy importante pues debe haber texto en las páginas por sentido común, para indicar cosas como son: títulos, instrucciones, artículos, y una infinidad de cosas más...

Para que un texto se muestre en nuestra página web simplemente lo introduciremos después de la etiqueta `<body>`, recordemos la estructura básica de HTML y con el siguiente ejemplo quedara claro (realiza el código):



Dicen que el `<title>` es, con mucha diferencia, lo más importante de una web. Lo usan los buscadores para indexar la página, pero (aún más importante) lo usamos las personas para saber en qué web entrar cuando un buscador muestra cientos de ellas. **Así que elige un buen título, y ni se te ocurra dejarlo en blanco.**



Vemos que entre la etiqueta **<title>** se encuentra el título de la página y entre **<body>** el texto que aparece en nuestra página web.



Práctica 1: Mi primera página web. Crea tu primer documento HTML como has aprendido hasta ahora, el título será “Mi primera página web” y en el cuerpo de la página escribe acerca de ti.

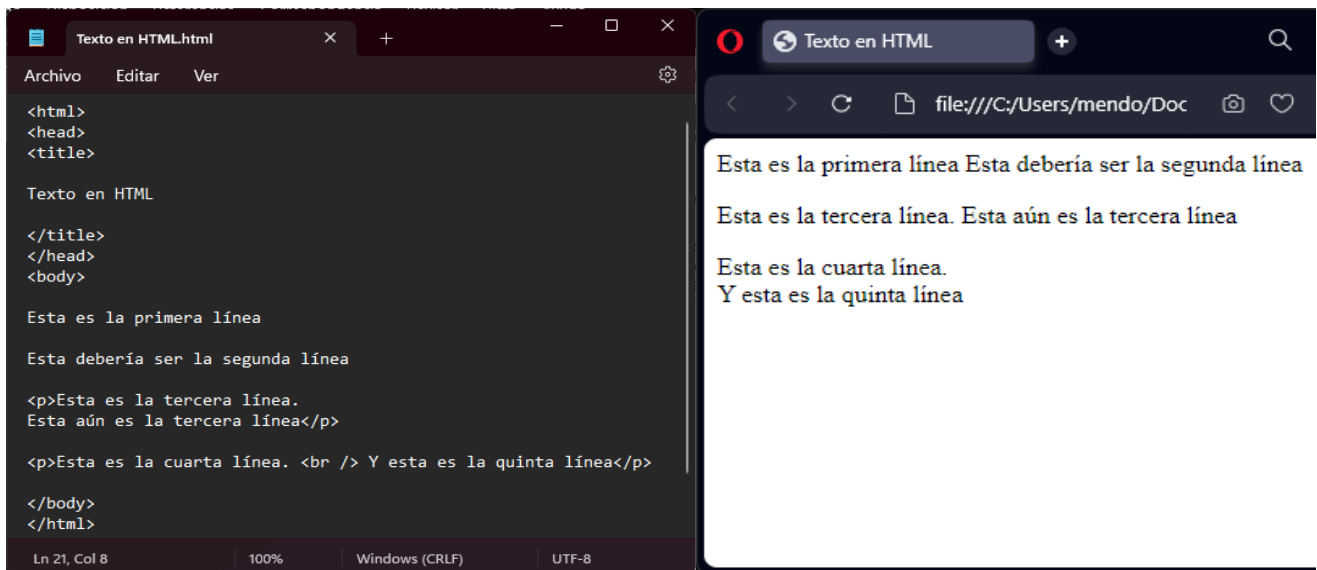


Recuerda: Al momento de hacer prácticas tendrás que hacer uso de todo lo que has aprendido y que conozcas, entonces si sabes cómo poner imagen de fondo, color, tipo de letra, color de letra o algún complemento que haga ver mejor tu trabajo también hará que la tu calificación mejore.

2.2 Párrafos y salto de línea

Un párrafo es un bloque de texto que se muestra separado del resto por un salto de línea.

Los párrafos se delimitan con las etiquetas **<p>** Con su respectivo cierre, es decir **</p>**. Si dentro del párrafo queremos hacer un salto o salto de línea, pero que se siga considerando un único párrafo (como dar un intro o enter en word) nosotros debemos escribir la etiqueta **
** donde queremos que del salto. Mira el siguiente ejemplo y realízalo también:



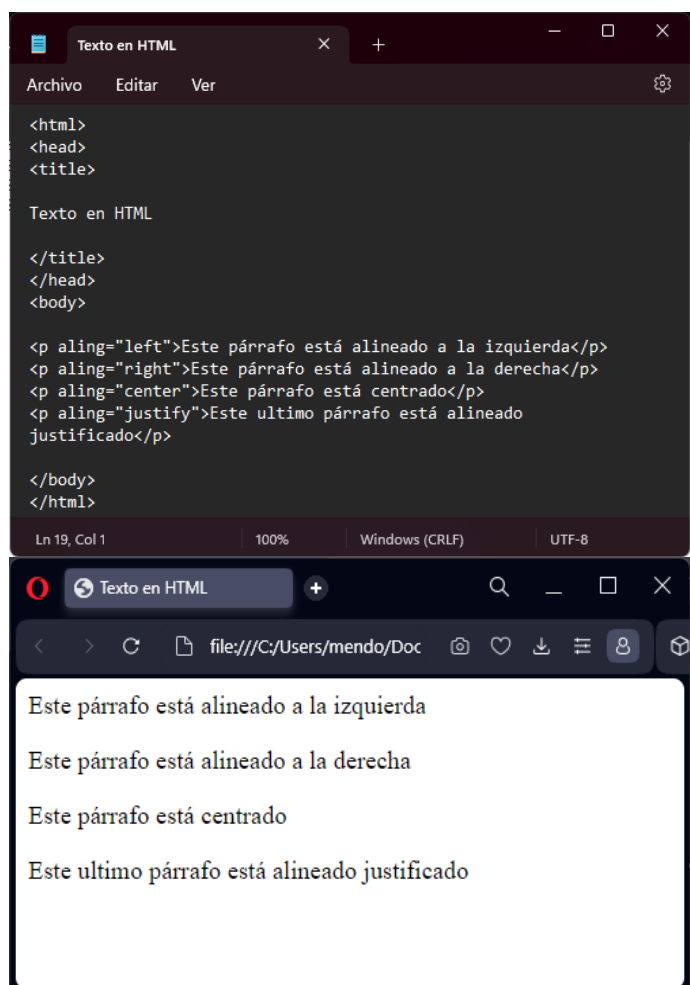
Podemos ver que HTML no diferencia los saltos de línea a menos que usemos un comando que le indique hacerlo.

Recuerda: A pesar de decir que cierres todas las etiquetas, no todas necesitan cierre, como la etiqueta `
` (salto de línea) pues esta no tiene cierre o la etiqueta `` que contiene auto-cierre.

2.3 Alineamiento de texto

Los párrafos creados con HTML son elementos de bloque, por lo que siempre ocupan toda la anchura de la ventana del navegador. Además, no tienen atributos específicos, pero sí que se les pueden asignar los atributos comunes de HTML básicos, de internacionalización y de eventos.

El alineamiento funciona para lograr acomodar la información de tu página en la dirección deseada. Es posible alinear elementos en bloque (tablas, imágenes, objetos, párrafos, etc.) en el lienzo con el atributo `<align>`. Aunque este atributo se puede establecer en muchos elementos HTML, su rango de valores posibles difiere en ocasiones de unos elementos a otros. Aquí sólo se considera el significado del atributo `<align>` para texto.



Los tipos de alineamiento son:

- ☑ **left**: las líneas de texto se representan con los bordes izquierdos alineados.
- ☑ **right**: las líneas de texto se representan con los bordes derechos alineados.
- ☑ **center**: las líneas de texto son centradas.
- ☑ **justify**: las líneas de texto se justifican en ambos márgenes.

Recuerda que **<align>** es un atributo por lo que irá dentro de la etiqueta principal como en el ejemplo (recrea el código del ejemplo).

Práctica 2: Refranes. Crea un nuevo archivo HTML, donde escribirás 4 refranes diferentes y cada uno con una diferente alineación.



2.4 Textos prediseñados

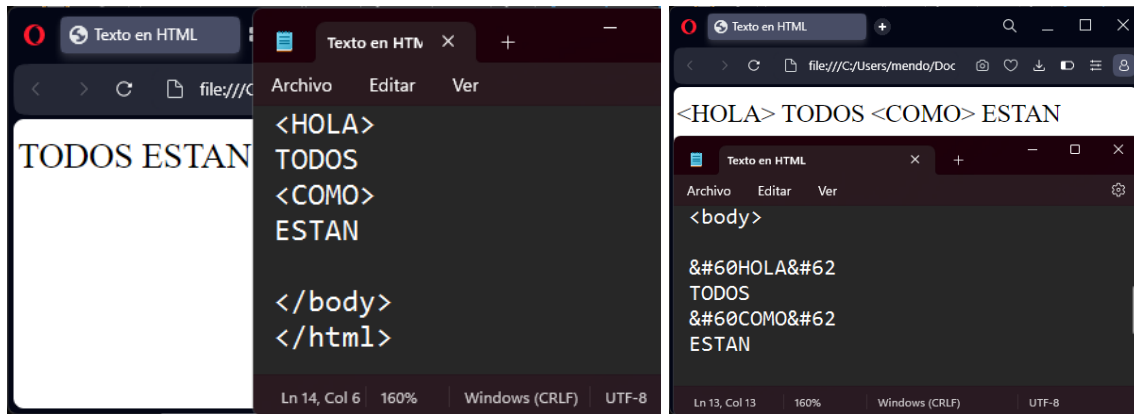
Como hemos visto en los trabajos anteriores, es necesario escribir algún comando que obligue a dar un salto de línea de lo contrario el navegador mostrara el texto de manera continua e ininterrumpida.

Pues a lo que está dentro de la etiqueta **<pre></pre>** le ocurre algo diferente, pues ésta etiqueta es “texto preformado” es decir que el formato de texto con el que escribamos lo va a respetar a como se visualiza en el editor de texto. Aquí el ejemplo (realiza el código del ejemplo):

```
<html>
<head>
<title>
  Texto en HTML
</title>
</head>
<body>
  <pre>
La
etiqueta      &#60pre&#62   Respeta
      el texto exactamente como
está
      escrito      C:
  </pre>
  <p>
Cosa
      que      no
      ocurre con
la etiqueta &#60p&#62   :c
  </p>
</body>
</html>
```

La
etiqueta <pre> Respeta
 el texto exactamente como
está
 escrito C:

Cosa que no
 ocurre con
la etiqueta <p> :c



Práctica 3: Texto predise. Crea un nuevo archivo HTML utilizando el texto prediseñado escribe como se hace una página web, recuerda anotar las etiquetas que se utilizan y sus funciones.



Práctica 4: Diferencias. Escribe en tus hojas de apuntes que diferencia tiene la etiqueta `<pre>` con las etiquetas `<p>` y `
`, además ¿cuál se te facilita?, ¿cuál utilizarías?, que ventajas y desventajas tiene cada una.



Recuerda: Los caracteres que utiliza HTML para definir sus etiquetas ("`<`" y "`>`") **no se pueden utilizar libremente**, ya que pueden ser interpretados de manera incorrecta por el navegador.

Consejo

Existe un sistema de codificación numérica que se utiliza para representar caracteres y símbolos en computadoras llamado código **ASCII**, podemos hacer de su uso utilizando los caracteres **&#** antes del código ASCII como se ve en el ejemplo. Solo a partir del número 30.

2.5 Negrita, Cursiva y Subrayado

Muchos procesadores de texto contienen esta función pues pese a ser simple nos brinda una gran ayuda a la hora de destacar texto de una manera sutil y no podemos dejar pasar por alto esta función así que en esta sección aprenderás a usarlo en HTML y veras lo simple que es.

Función	Etiqueta	Ejemplo	Resultado
Negrita (Bold)	<code>...</code>	La palabra está en <code></code> negritas <code></code>	La palabra está en negritas
Cursiva (Italic)	<code><i>...</i></code>	<code><i></code> Todo está en cursiva <code></i></code>	<i>Todo está en cursiva</i>
Subrayado (Underline)	<code><u>...</u></code>	Lo <code><u></code> encerrado <code></u></code> está <code><u></code> subrayado <code></u></code>	Lo <u>encerrado</u> está <u>subrayado</u>

Ahora si recordamos la sintaxis de HTML recordaremos que si queremos modificar una parte del texto basta con ponerlo entre dos etiquetas con una función específica, que en este caso serán:

Práctica 5: Vocales.

Transcribe el texto en un nuevo archivo HTML entonces pon en negritas todas las palabras con E, subraya todas las palabras con A y en cursiva todas las que contengan alguna O.



Explosión - Delmira Agustini

Si la vida es amor, ¡bendita sea!

¡Quiero más vida para amar! Hoy siento

Que no valen mil años de la idea

Lo que un minuto azul de sentimiento.

Mi corazón moría triste y lento...

Hoy abre en luz como una flor febea;

¡La vida brota como un mar violento

donde la mano del amor golpea!

Hoy partió hacia la noche, triste, fría,

Rotas las alas, mi melancolía;

Como una vieja mancha de dolor

En la sombra lejana se deslíe...

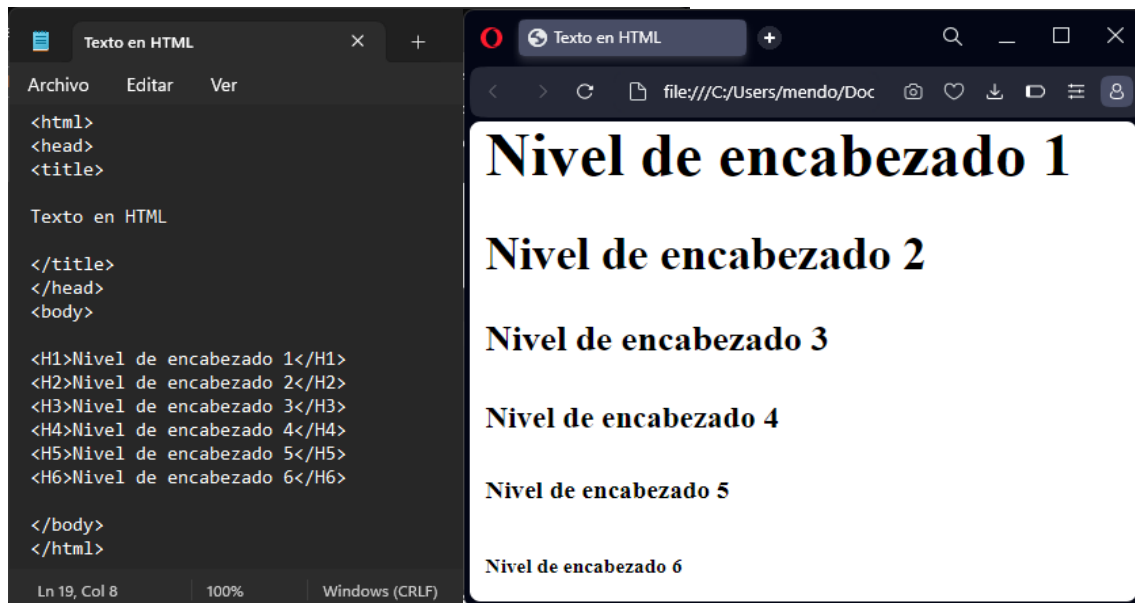
¡Mi vida toda canta, besa, ríe!

¡Mi vida toda es una boca en flor!

2.6 Títulos

Como la mayoría de los documentos y páginas web necesitamos dar una buena impresión al usuario, una herramienta que nos puede servir para captar la atención es el título o en casos los subtítulos. Los “Headings” son etiquetas que definen los títulos y se pueden utilizar para marcar el título y/o los diferentes apartados que forman el documento, o simplemente para resaltar alguna parte del texto. Aun con eso recuerda no exagerar su uso, ya que los “Headings” son para un uso más estratégico usándolos para remarcar palabras clave.

Headings viene del inglés y significa encabezados. Existen seis tipos de headings, que van de la etiqueta `<h1>` (la más grande) a `<h6>` (la más pequeña). Fíjate en el ejemplo y replícalo:



Práctica 6: Escribe tu nombre. Crea un archivo HTML con la estructura básica y el título “Encabezados 1”. Escribe tu nombre con cada tipo de encabezado.



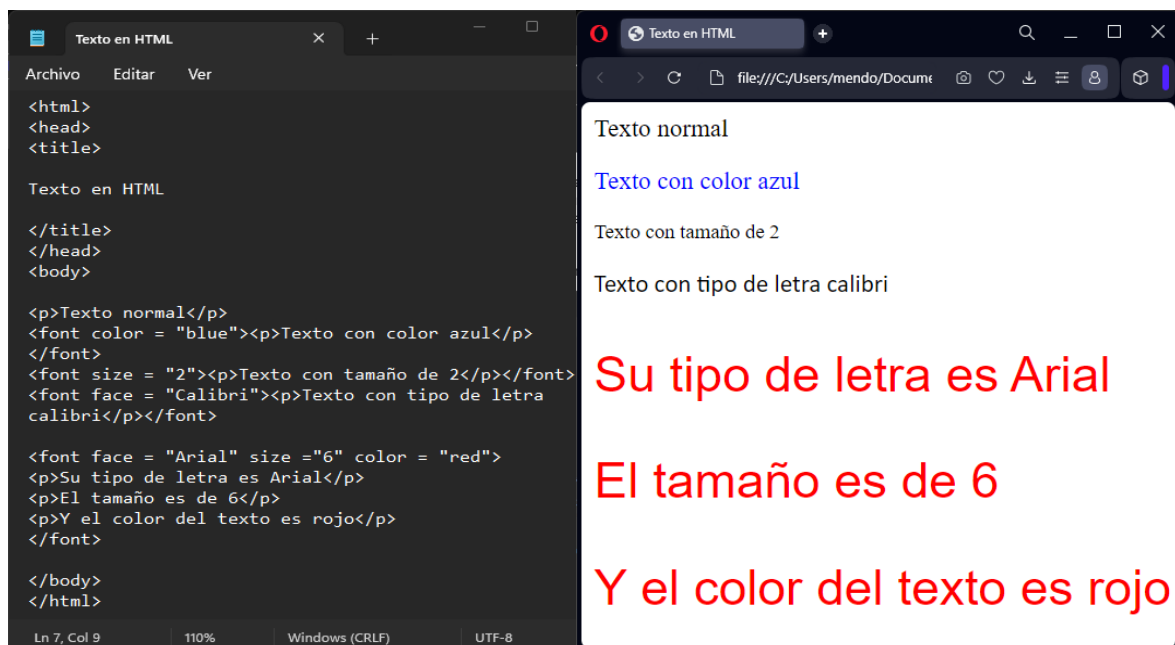
He de remarcar que los headings son para títulos subtítulos y palabras clave, **no para cambiar el tamaño de letra de la página.**

2.7 Fuente de texto

Hasta ahora el texto de HTML casi siempre ha tenido el mismo tipo de fuente, el mismo tamaño y el mismo color, pero eso cambia gracias a la etiqueta `...`, pero esta etiqueta no hace mucho por sí sola, si no viene acompañada de alguno de los siguientes atributos:

- AA Size: Nos permite cambiar el tamaño de la fuente a partir de un número siendo 1 la más pequeña y 7 la más grande posible.
- AA Color: este atributo nos permite cambiar el color de la fuente mediante números hexadecimales, código RGB o uno que otro color en idioma inglés.
- AA Face: Para tener algún tipo de fuente diferente al predeterminado por el navegador es necesario utilizar este atributo.

El siguiente ejemplo lo ilustra mejor:



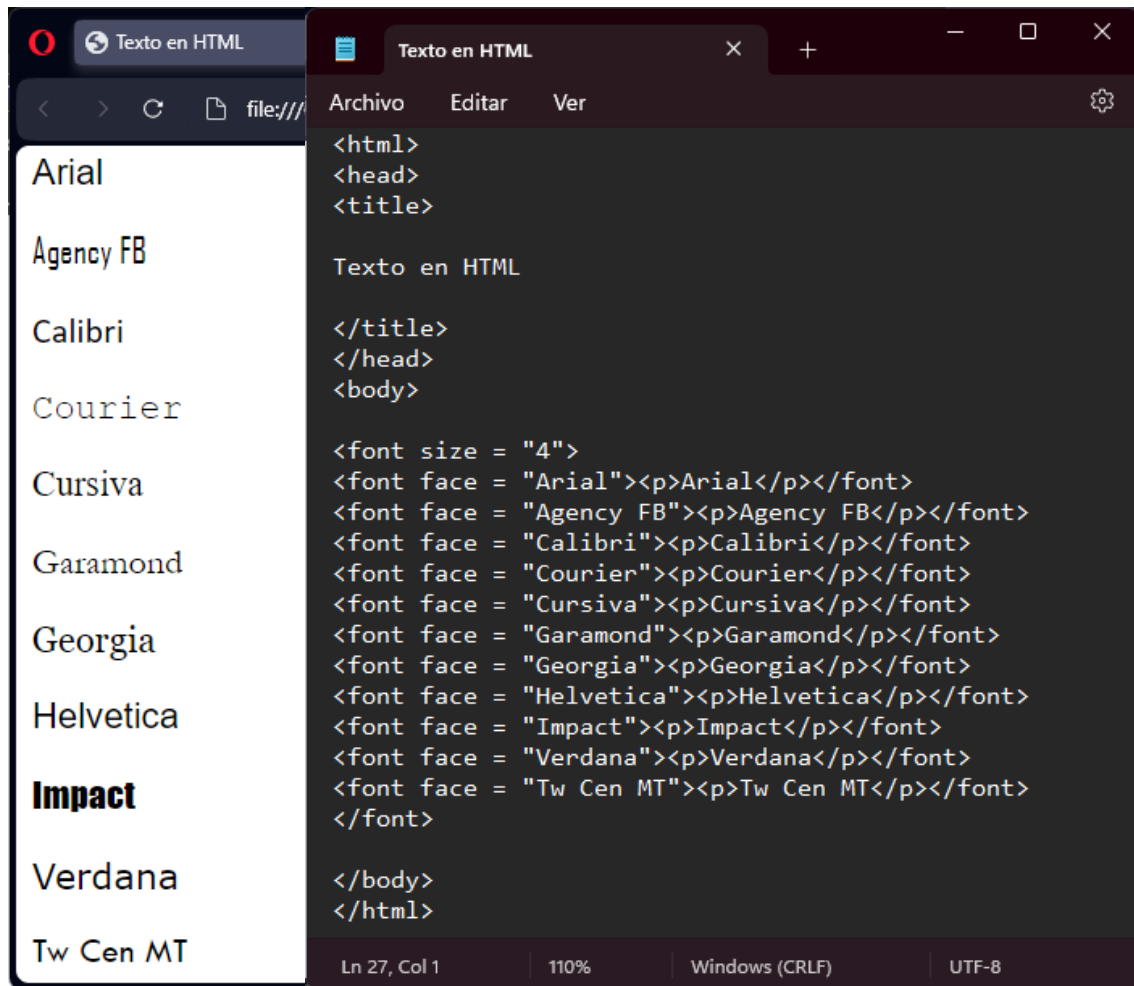
Práctica 7: Colores. Investiga cuales son 140 nombres de los colores que soporta HTML y anota al menos 10 de ellos con su nombre, su código hexadecimal y el código RGB.



Consejo

En este manual se enseña lo más **esencial** del lenguaje de etiquetas, habrán atributos o etiquetas que no se presentan aquí, pero que bien puedes buscar en internet, no solo mejorara tu trabajo si no que te mejorara tus habilidades autodidactas.

Hay que resaltar que el tipo de fuente dependerá tanto del sistema operativo y de las fuentes que tu navegador tenga instalado. Aquí unos ejemplos de fuentes (realiza el ejemplo):



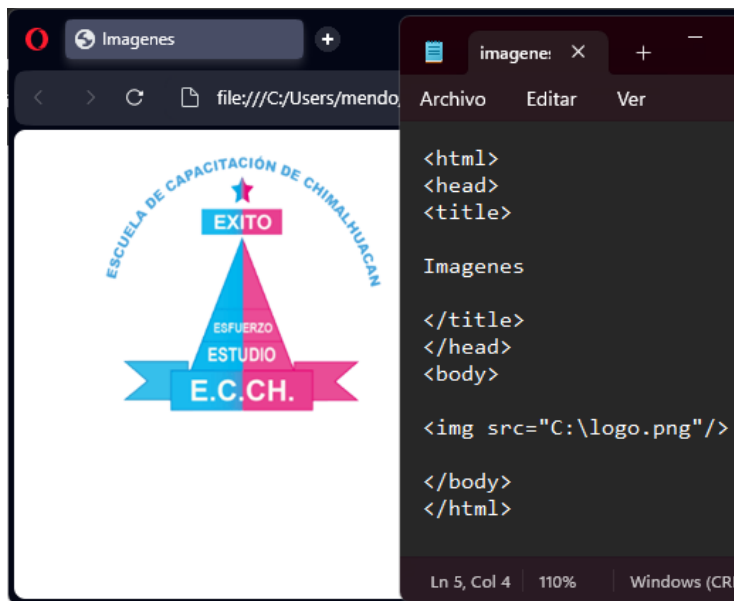
Práctica 8: Poemas. Escribe 2 poemas cada uno con fuentes, colores y tamaños diferentes (Recuerda que la creatividad es lo más importante).



3. Imágenes

Uno de los elementos más importantes de HTML son las imágenes, las páginas web actuales contienen al menos una imagen, mientras otras páginas constan de docenas de imágenes, pues este es un recurso más llamativo y enriquecedor que el texto.

La manera de insertarlas no es de otro mundo, pues solo hay que agregar la etiqueta ****, la etiqueta **** es la correspondiente a imagen, el atributo **"src"** (del inglés "source" en español "fuente") es obligatorio y es el que nos dirá en donde se encuentra alojada la imagen, en otras palabras, la dirección donde se encuentra guardada la imagen, "x" es la dirección del archivo o URL dependiendo el



caso y por ultimo como **** es una etiqueta sin cierre tendremos que colocar "/" antes de cerrar. Con siguiente ejemplo será más fácil de comprenderlo (realízalo):

Recuerda: En donde va a ir la dirección del archivo habrá que escribir el nombre del documento y al final la extensión, es decir: .JPG, .PNG o .GIF

Consejo

Mantén los archivos que vas a utilizar, en la carpeta donde esta tu archivo HTML, ya que esto te evitara escribir la ubicación o siempre puedes utilizar direcciones URL.

3.1 Tamaño de la imagen

A la hora de descargar una imagen está siempre tendrá un tamaño predeterminado y si es una imagen con alto detalle o de alta definición lo más seguro es que la imagen sea muy grande y en el momento de utilizarlo en tu página web llega a utilizar mucho (si no es que todo) el espacio del lienzo o en ocasiones nos puede pasar todo lo contrario y encontrar una imagen que apenas es visible a pesar de lo simple de la imagen.

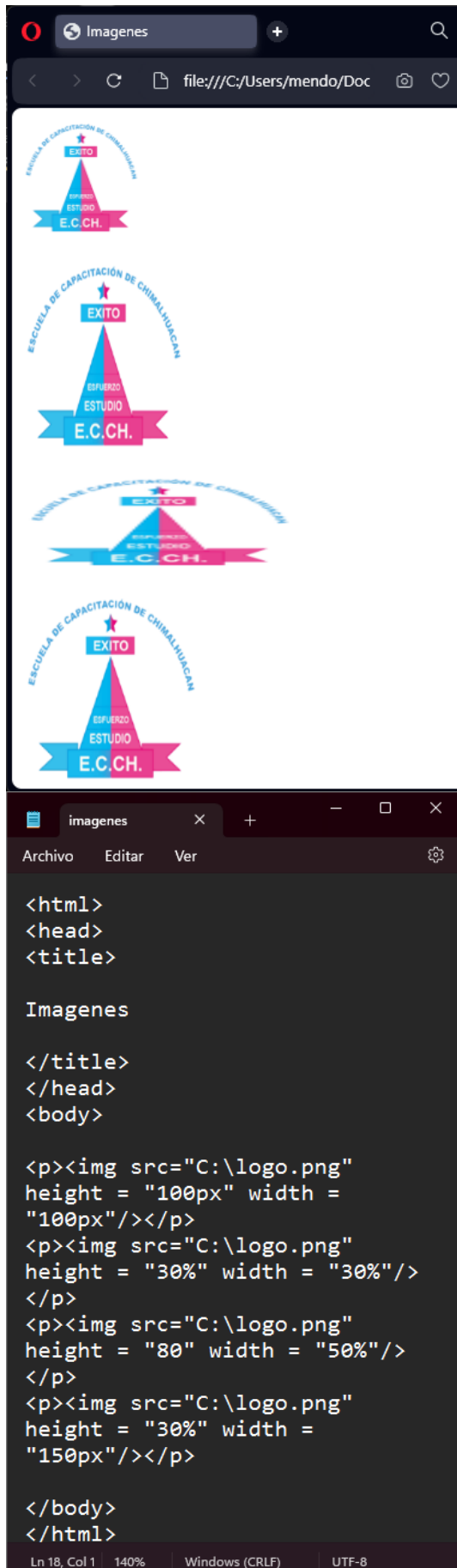
Afortunadamente podemos evitar esto con los atributos de tamaño:

- ☐ Height: este atributo modifica la altura
- ☐ Width: este atributo modifica lo ancho

Ambos atributos tienen que ser descritos con valores numéricos ya sea en píxeles (px) o en porcentaje (%) y si no se especifica el valor será tomado en píxeles. Mira el siguiente ejemplo y hazlo tú también (puedes usar la imagen que sea de tu agrado):

Los valores en porcentaje serán relativos al tamaño de la ventana mientras que los píxeles serán absolutos.





Practica 1: Inserta 10 imágenes.

Crea un nuevo archivo HTML con 10 imágenes y con tamaños diferentes.



3.2 Alineación

Ya hemos visto el atributo **"align"** y conocemos como funciona, pues con las imágenes es exactamente igual. Ejemplo:

```
<img src = "logo.png" align = "center">
```

Practica 2: Noticia. Crea un nuevo archivo HTML y escribe una noticia, con título, imágenes, y la nota.



3.3 Imagen de fondo

En una parte de este manual hemos visto de manera muy breve como agregar color de fondo, a la página web, el agregar una imagen no es muy diferente, a continuación, veras la manera y la diferencia de usar color de fondo.

Imagen

```
<body background = "imagen.jpg">
```

Color

```
<body bgcolor = "blue">
```

Como te puedes dar cuenta es en la etiqueta `<body>` y dependiendo el tamaño de la imagen es como se mostrará en la página, si la imagen es muy chica aparecerá en mosaico, y si es muy grande cubrirá toda la página, en caso de solo poner el color tendremos que utilizar los colores en inglés o con hexadecimales como vimos en el color de las letras.

4. Marquesinas

Las marquesinas son renglones usualmente de texto (OJO, esto no indica que no se pueda usar en otros objetos como imágenes), que se desplazan de un punto a otro, esto para dar un mejor efecto a la página y que no es difícil de conseguir.

Práctica 1: Ejemplo de marquesina. Crea un nuevo archivo HTML con la estructura básica y transcribe el siguiente texto, después de la etiqueta body, anota en tus hojas de apuntes las etiquetas o atributos que reconoces, las etiquetas que no conozcas, solo escríbelas, las conocerás después.

```
<marquee bgcolor= "blue" direction= "right" >  
<b><font color= "yellow" size= "5" > Esto es una marquesina </font></b>  
</marquee>
```



6.1 Direcciones de marquesinas

Lo que acabas de ver es una marquesina, estas se activan con la etiqueta `<marquee>` y su respectivo cierre `</marquee>`, cuando escribimos la marquesina por si sola la dirección predeterminada será hacia la izquierda, pero gracias al atributo "direction" podremos cambiarlo, solo tendremos que poner alguna dirección en inglés, como son: left (izquierda), right (derecha), up (arriba) y down (abajo).

Como en la práctica anterior, la dirección establecida ha sido: right (derecha).

Junto a las direcciones de la marquesina podemos encontrar dos atributos que ayudan a la animación del movimiento y esas son:

loop: Este atributo nos determina si el movimiento se repetirá o no. Si queremos que se repita sólo 5 veces, su loop será el siguiente: `<marquee loop= "5"> </marquee>`

behavior: Con este atributo podemos dar nuevos efectos a la marquesina, con `behavior="slide"` haremos que el texto se detenga al llegar al final de la marquesina. Y con el `behavior="alternate"` el texto irá y volverá de un lado a otro de la marquesina.

Práctica 2: Diferentes marquesinas. Crea un nuevo archivo HTML y escribe 4 diferentes marquesinas con las diferentes direcciones, agregando loop o behavior, con diferentes colores de fondo y diferentes colores en letras.



Los atributos **height** y **width** no solo modifican el tamaño de las imágenes, también pueden modificar: tablas, videos, cuadros de texto, etc... en ese caso puede modificar el tamaño de la marquesina.

6.2 Velocidades de marquesinas

Aquí podremos utilizar dos atributos los cuales serán:

scrollamount: Este atributo define la cantidad de píxeles que queremos que se desplace el texto en cada movimiento. Por ejemplo: `<marquee scrollamount="3">`
`</marquee>` significará que el texto que vaya dentro de esa etiqueta se moverá a razón de tres píxeles por movimiento. Mientras mayor sea el número más rápido ira.

scrolldelay: Éste nos define la velocidad del texto que está dentro de la marquesina. A menor numeración, mayor velocidad. Es decir, un texto irá más rápido si el scrolldelay es 5, que si el scrolldelay es 500. Esta etiqueta nos ayudara para hacer que la marquesina vaya a una velocidad más lenta.

Práctica 3: 10 imágenes. Inserta 10 imágenes diferentes y crea marquesinas con ellos recuerda emplear todo lo que ya has visto.



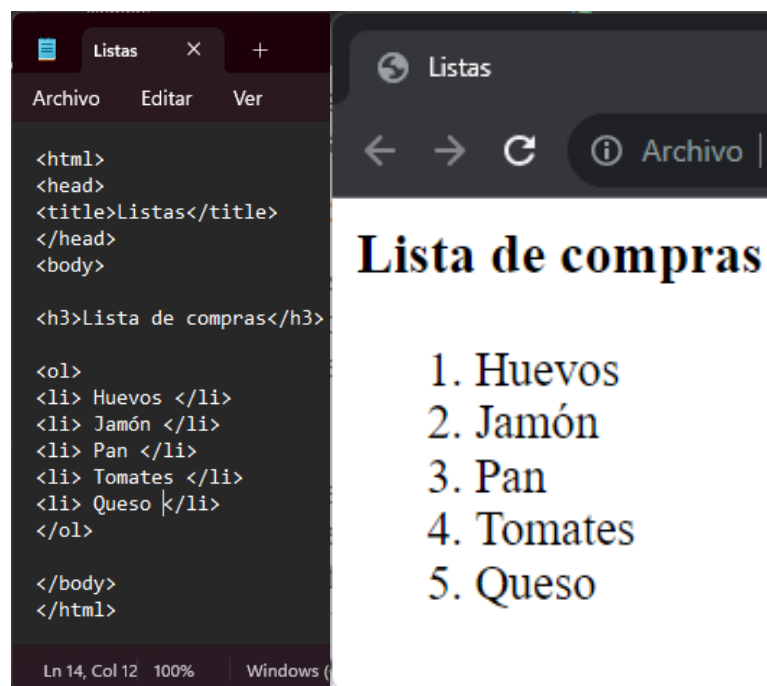
5. Listas

En HTML Existen tres tipos de listas diferentes, listas ordenadas, no ordenadas y de definición, cada una con cualidades que nos ayudaran a realizar una tarea diferente, pero que se pueden utilizar juntas para enriquecer nuestro trabajo.

Las listas no son diferentes a las que solemos utilizar como en Word, pero si un poco más limitantes en el aspecto gráfico y visual.

5.1 Listas Ordenadas

Las listas ordenadas son listas enumeradas, para hacer uso de este tipo de listas es necesario abrir con la etiqueta `` esto dictara a la página web que iniciara un apartado de listas ordenadas, después de esto es necesario escribir `` esto indica que lo que siga después es un artículo de la lista, puedes poner tantos `` como necesites para realizar tu lista enumerada solo recuerda cerrar la etiqueta ``. Observa el siguiente ejemplo y replicalo:



Práctica 1: Canciones. Crea una lista ordenada, con 10 elementos de tus canciones favoritas. Si queremos que nuestra lista empiece por el número 10, sólo deberemos escribir lo siguiente:

`<li value = "10"> Iniciara en el 10 `



5.2 Listas No ordenadas

Las listas no ordenadas son las que usualmente contienen viñetas o figuras, pues como su nombre lo dice no tiene algo que indique un orden solo te muestras que es una lista. A diferencia de las listas ordenadas la etiqueta de inicio en este caso será `` pero la etiqueta de los objetos continuará siendo ``.

¿Notas la diferencia?

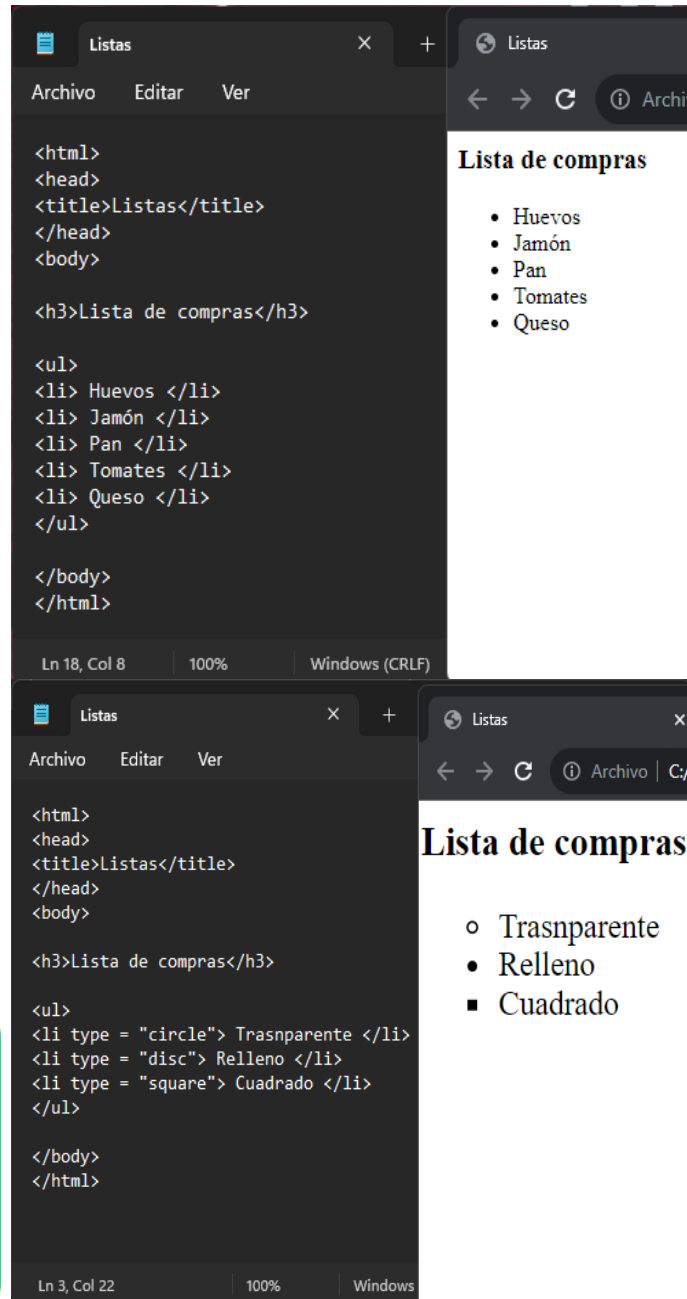
★ Recuerda hacer apuntes ★

En HTML esta "viñeta" puede ser un círculo oscuro, un círculo transparente o un cuadrado oscuro, esto lo modificas mediante el atributo `"type"`. Si no escribes el atributo `"type"` la viñeta predeterminada será el círculo relleno o "disc".

Observa el ejemplo siguiente y realízalo:

Práctica 2: Reglas del salón.

Crea una lista no ordenada, en donde escribirás 10 cosas que debes evitar en clase.

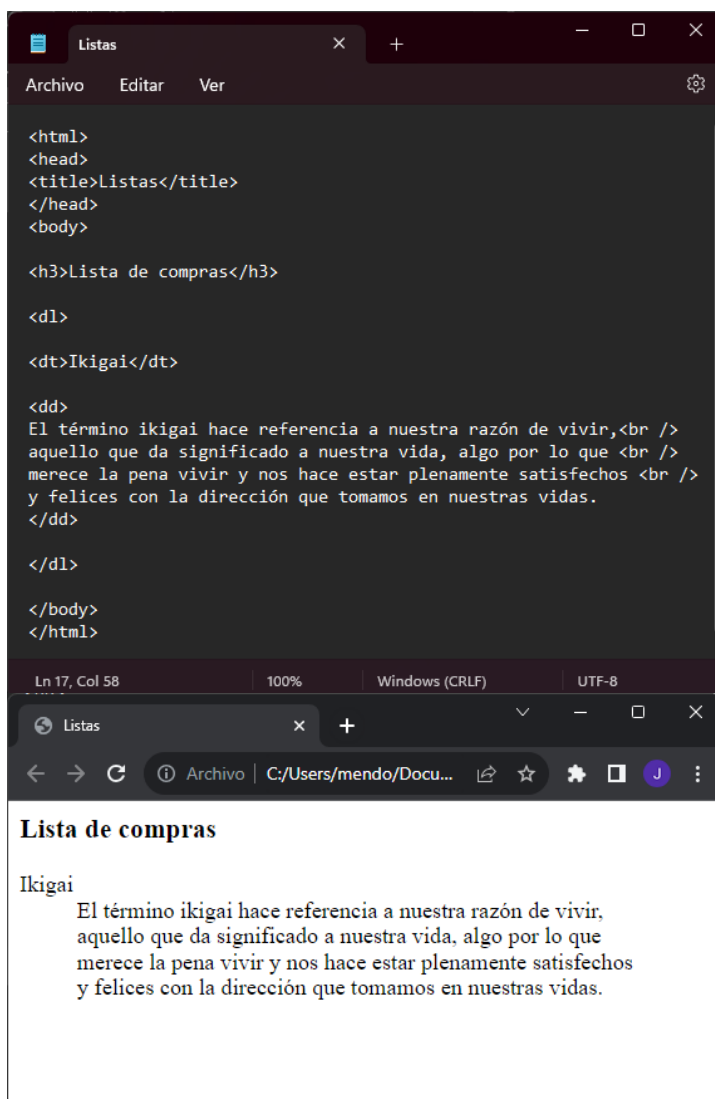


5.3 Listas De definición

Las listas de definición no son listas continuas, más bien son listas multinivel, esto significa que mantendrán una jerarquía y al primer punto se le dará mayor relevancia que al segundo. Es de definición ya que es utilizada de forma habitual para dar una breve explicación de la palabra título por así decirle.

Se diferencia por utilizar tres etiquetas en lugar de dos, para iniciar la lista es necesario escribir la etiqueta `<dl>` para escribir la palabra título se pondrá la etiqueta `<dt>` y para definirla se utilizará `<dd>` hay que recordar darles sus respectivos cierres pues, aunque a veces no es necesario, la sintaxis y las normas indican que hay que hacerlo, a menos claro que estas mismas digan que no lo lleve.

Este tema será más fácil de entender con el siguiente ejemplo (recreálo):



The image shows a code editor window titled 'Listas' with the following HTML code:

```
<html>
<head>
<title>Listas</title>
</head>
<body>

<h3>Lista de compras</h3>

<dl>

<dt>Ikigai</dt>

<dd>
El término ikigai hace referencia a nuestra razón de vivir,<br />
aquello que da significado a nuestra vida, algo por lo que <br />
merece la pena vivir y nos hace estar plenamente satisfechos <br />
y felices con la dirección que tomamos en nuestras vidas.
</dd>

</dl>

</body>
</html>
```

Below the code editor, a web browser window shows the rendered output. The title is 'Lista de compras' and the content is a definition list for 'Ikigai'.

Lista de compras

Ikigai
El término ikigai hace referencia a nuestra razón de vivir, aquello que da significado a nuestra vida, algo por lo que merece la pena vivir y nos hace estar plenamente satisfechos y felices con la dirección que tomamos en nuestras vidas.

Práctica 3: Receta. Crea un nuevo archivo HTML y en este escribirás una receta (la que tú quieras), esta deberá de contener:

1. El nombre
2. Descripción del platillo
3. Una imagen del platillo
4. Los ingredientes
5. Los pasos a seguir

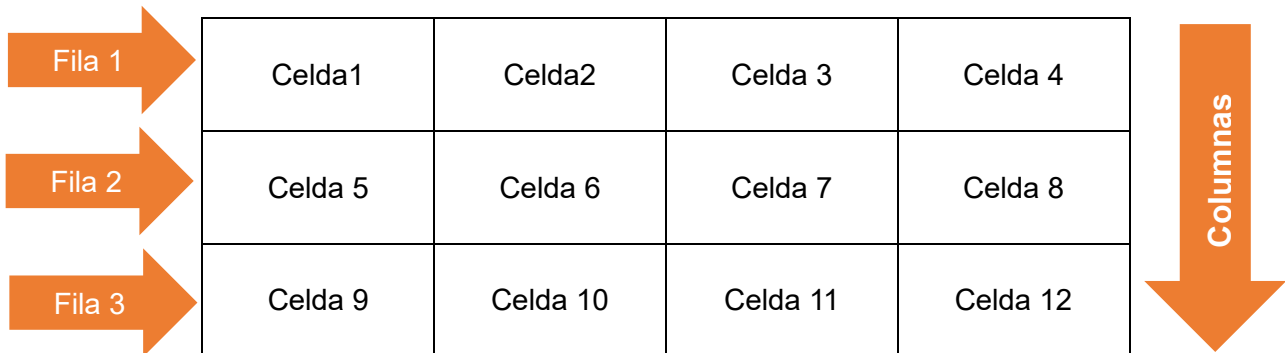


Consejo

Ya dijimos que hay que cerrar todas las etiquetas a menos que se indique lo contrario. Tal vez ya te diste cuenta pero hay veces en las que abrimos una nueva etiqueta sin cerrar la anterior y al ver la página no da problemas. Esto se debe a que muchas de las etiquetas de cierre son **OPCIONALES**. Aun así evita los malentendidos y ciérralo por tu cuenta, ya que muestra limpieza y un correcto uso del lenguaje.

6. Tablas

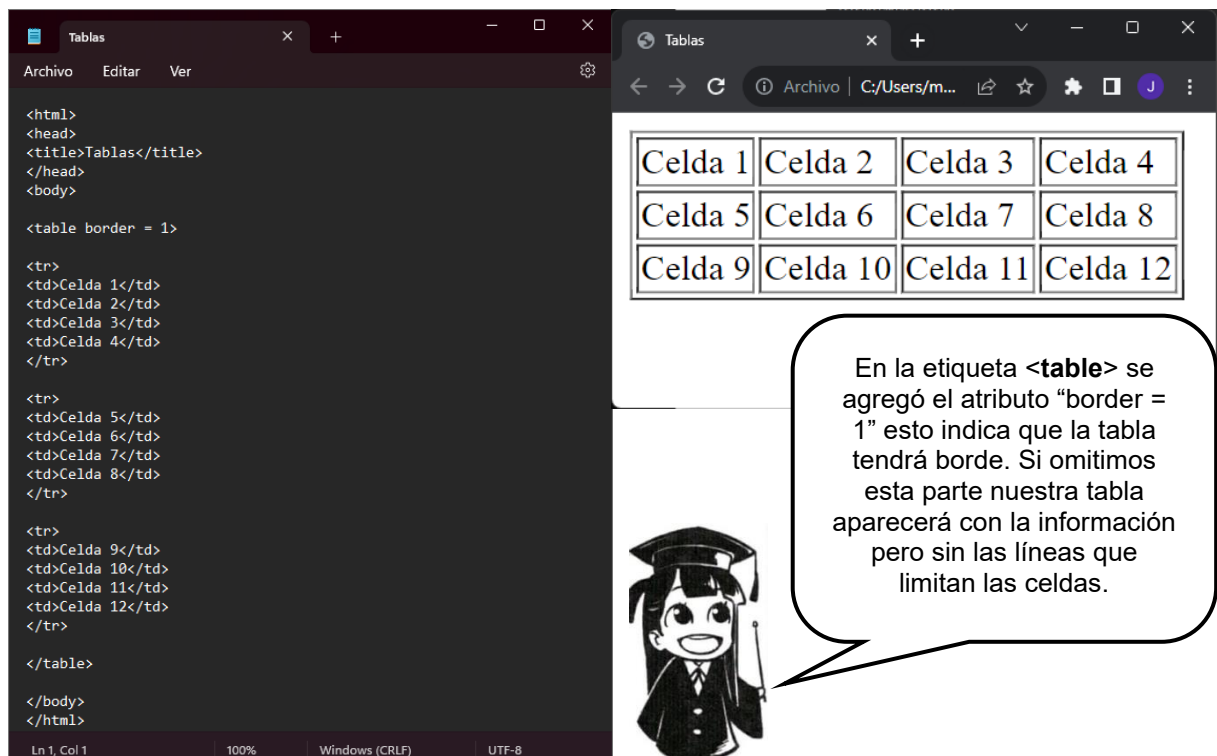
Hasta este punto conocemos que es una tabla, resumido no es otra cosa más que un medio de organizar datos en filas y columnas. Así que nos enfocaremos en la estructura de estas con el lenguaje HTML, habrá que empezar aclarando que para realizar una tabla debemos tener en cuenta que se está realizando de izquierda a derecha y de arriba abajo. Es decir, empezamos desarrollando la primera fila, dentro de ella, las determinadas columnas que haya, y a continuación, la siguiente fila.



The diagram illustrates a 3x4 table structure. On the left, three orange arrows point to the rows, labeled 'Fila 1', 'Fila 2', and 'Fila 3'. On the right, a large orange arrow points downwards, labeled 'Columnas'. The table itself contains 12 cells, numbered 'Celda 1' through 'Celda 12' in a sequential order from top-left to bottom-right.

Celda 1	Celda 2	Celda 3	Celda 4
Celda 5	Celda 6	Celda 7	Celda 8
Celda 9	Celda 10	Celda 11	Celda 12

Ya teniendo claro esto continuaremos explicando cuales son las etiquetas que utilizaremos. Iniciaremos con la etiqueta de apertura que será **<table>** esto le indica a la página web que en el siguiente apartado se generara una tabla, entonces ya sabiendo esto utilizaremos la **<tr>** esta etiqueta indica que a partir de aquí se generara una fila, al final pondremos la etiqueta donde podremos empezar a llenar la tabla esta es **<td>**, con esta etiqueta indicaremos que se creara una celda, si ponemos otra **<td>** creara la celda número 2 de la fila 1, si insertamos otro **<td>** creara la celda número 3 de la fila 1 así sucesivamente hasta que cerremos el **<tr>**, si queremos crear la fila número 2 volveremos a escribir **<tr>** para indicar que crearemos otra fila y lo empezaremos a llenar de **<td>** para crear más celdas. Este ejemplo muestra lo antes descrito (realízalo):



The image shows a side-by-side comparison of HTML code and its rendered output. On the left, a code editor window titled 'Tablas' displays the following HTML code:

```
<html>
<head>
<title>Tablas</title>
</head>
<body>

<table border = 1>

<tr>
<td>Celda 1</td>
<td>Celda 2</td>
<td>Celda 3</td>
<td>Celda 4</td>
</tr>

<tr>
<td>Celda 5</td>
<td>Celda 6</td>
<td>Celda 7</td>
<td>Celda 8</td>
</tr>

<tr>
<td>Celda 9</td>
<td>Celda 10</td>
<td>Celda 11</td>
<td>Celda 12</td>
</tr>

</table>

</body>
</html>
```

On the right, a browser window titled 'Tablas' shows the rendered table, which matches the one in the diagram above. A speech bubble from a cartoon character explains the 'border' attribute.

En la etiqueta **<table>** se agregó el atributo "border = 1" esto indica que la tabla tendrá borde. Si omitimos esta parte nuestra tabla aparecerá con la información pero sin las líneas que limitan las celdas.

Práctica 1: Tabla 1. Crea un nuevo archivo HTML y crea la estructura para hacer la siguiente tabla.



Horario de clases

Horas	Lunes	Martes	Miércoles	Jueves	Viernes
8:00 - 8:50	Física	Historia	Ed. Física	Química	Dibujo
8:50 - 9:40	Historia	Matemáticas	Lengua	Sociales	Naturales
9:40 - 10:30	Dibujo	Historia	Dibujo	Ed. Física	Sociales
10:30 - 10:45			Recreo		
10:45 - 11:35	Naturales	Inglés	Lengua	Música	Física
11:35 - 12:25	Historia	Química	Sociales	Matemáticas	Filosofía
12:25 - 13:15	Filosofía	Física	Química	Dibujo	Inglés

Práctica 2: Tabla 2. Crea un nuevo archivo HTML, y estructura el código para realizar la tabla y el contenido como se muestra en la imagen, recuerda modificar la fuente de la letra y color de fondo.



PANES	DESCRIPCIÓN	PRECIO
Chapata	Ligeramente tostada y hecha con harina de trigo	\$35
Trigo mixto	Pan integral suave con grano mixto	\$80
Pan campestre	Extremadamente suave y tierno	\$90
Brioche	Pan denso y sabroso, hecho con huevos orgánicos y leche	\$100
Pan especial	Hecho con harina de garbanzo y un toque de romero	\$110

6.1 Combinar celdas

Para combinar varias columnas o filas utilizamos los atributos **colspan** y **rowspan** respectivamente.

Colspan: nos permite unir celdas de manera horizontal, este atributo requiere un valor numérico para funcionar, este número será el de las celdas que queramos combinar.

Rowspan: este atributo nos permite unir celdas verticalmente, de la misma manera requiere un valor numérico.

Consejo

El manejar estos atributos en ocasiones llega a ser confuso, así que se recomienda crear un boceto de la tabla en otro lugar como en papel para lograr una mejor visualización de lo que se quiere lograr en HTML.

Si quieres destacar una fila o columna para que se comporte como título utilizamos **<th>...</th>**. Mira los siguientes ejemplos y realízalos:

The screenshot shows a code editor with the following HTML code:

```
<html>
<head>
<title>Tablas</title>
</head>
<body>

<table border = 1>

<tr>
<th colspan = 3>Uso de Colspan</th>
</tr>

<tr>
<td colspan = 3 align = "Center">Celda 1, 2 y 3</td>
</tr>

<tr>
<td>Celda 5</td>
<td>Celda 6</td>
<td>Celda 7</td>
</tr>
</table>

</body>
</html>
```

Below the code editor, the rendered table is displayed:

Uso de Colspan		
Celda 1, 2 y 3		
Celda 5	Celda 6	Celda 7

The screenshot shows a code editor with the following HTML code:

```
<html>
<head>
<title>Tablas</title>
</head>
<body>

<table border = 1>
<th colspan = 3>Uso de Rowspan</th>

<tr>
<td rowspan = 2>Celda 1 y 5</td>
<td>Celda 2</td>
<td rowspan = 2>Celda 3 y 7</td>
</tr>

<tr>
<td>Celda 6</td>
</tr>
</table>

</body>
</html>
```

Below the code editor, the rendered table is displayed:

Uso de Rowspan		
Celda 1 y 5	Celda 2	Celda 3 y 7
	Celda 6	

Práctica 3: Tabla 3. Crea un nuevo archivo HTML y desarrolla el código para realizar la siguiente tabla.



DIFERENCIAS ENTRE EL PERRO Y EL HOMBRE			
Diferencias	Perro		Hombre
	Pequeño	Grande	
Duración crecimiento	10 meses	18 a 24 meses	16 años
Tiempo de gestación	58 a 63 días		9 meses
Duración de vida del pelo/cabello	1 año		2 a 7 años

7. Enlaces

Los enlaces nos permiten acceder a otras páginas, sitios web, ficheros (de música, vídeos, programas, imágenes, etc.), permiten descargar archivos de la red y acceder a otras partes o secciones dentro de un documento, la etiqueta para utilizar un enlace no es `<a>` con el atributo “href” indicando la dirección de la página, al final quedara así:

```
<a href = "http://www.google.com">Da clic aqui</a>
```

Si el enlace va dirigido a un archivo que tú hayas creado en lugar de utilizar un URL pondremos el nombre del archivo más su extensión, como si de una imagen se tratase, en su lugar quedara así:

```
<a href = "El nombre del archivo.html">Da clic aquí</a>
```

Aquí debo mencionar un atributo muy importante: `target="blank"`, para que el enlace se abra en una pestaña nueva del navegador.

Recuerda: Todo lo que este entre la etiqueta de apertura y la de cierre se ve afectado por estas.

Práctica 1: 5 historias. Crea un cinco nuevos archivos HTML y en cada uno de ellos escribe una historia diferente con una imagen relacionada a este, cada uno enlazado a otro archivo, esto quiere decir que en alguna parte de tu documento habrá una imagen que te lleve a la siguiente historia.



Practica 2: Índice. Crea un archivo HTML índice que te lleve a cada uno de los trabajos que has hecho a lo largo del manual (mínimo 1 por tema).



8. Audio

La etiqueta audio es un elemento que agrega gran vida a HTML, es sumamente sencillo y con un gran potencial a explotar, en este manual se te presentara la manera más sencilla y eficaz de introducir audio a tus páginas. La estructura de audio, se parece mucho a la de imagen pues esta aparte de no tener cierre, necesita del atributo “src” para funcionar. La etiqueta principal de audio es <audio> (verdad que no es muy difícil), entonces esto quedaría de la siguiente manera:

```
<audio src = “Nombre del archivo.mp3” />
```

Como te das cuenta es muy parecido a insertar una imagen, pero con archivos totalmente diferentes, nuevamente recomendamos que tu archivo de sonido se encuentre en el mismo lugar que tu documento HTML para poder agilizar y evitar el tener que escribir el directorio de la imagen.

En segundo plano tenemos que hablar de los formatos soportados por los navegadores en cuanto a archivos de sonido hablamos.

Formato	Codec utilizado	Características	¿Recomendado?
MP3	MPEG Layer-3	Buena calidad.	Sí, buen soporte
OGG	Ogg Vorbis	Buena calidad. Alternativa libre a MP3.	Sí, soporte medio
AAC	Advanced Audio Coding	Mejora el MP3. Usado como audio en MP4.	Sí, soporte medio
Opus	Opus	Buena calidad. Alternativa libre a MP3.	Sí, soporte malo
FLAC	FLAC Audio Lossless	Compresión sin pérdidas. Alto tamaño.	Sí
WAV	Wave sound	Formato de Microsoft. Está soportado.	No, muy pesado

Sin embargo, esto no mostrará nada visualmente, ni reproducirá ningún sonido. Realmente, el audio está ahí, pero no le hemos indicado el atributo “controls” para que muestre los controles y que el usuario pueda pulsar el botón “play”. Tampoco le hemos colocado el atributo “autoplay” que reproduce el audio automáticamente, en cuyo caso tampoco mostraría nada visualmente, pero sí que se empezaría a escuchar el sonido o pista de música. Los resultados correspondientes son los siguientes:

```
<audio src = "nombre del archivo.mp3" controls />
```

```
<audio src = "nombre del archivo.mp3" autoplay />
```

Práctica 1: Canción con letra. Crea un nuevo archivo HTML e inserta una canción junto con la letra de esta.



Para terminar, podremos agregar el atributo “loop”, ya con este atributo se reiniciará el audio cuando termine de reproducirse. Al final podrá quedar de la siguiente manera:

```
<audio src = "nombre del archivo.mp3" autoplay loop />
```

Esto reproducirá de manera automática la canción y cuando esta termine se reiniciará automáticamente.

Recuerda: Estas etiquetas se pueden combinar.

Práctica 2: Rockola. Crea un nuevo archivo HTML y en este crearás una rockola con un mínimo de 5 canciones, recuerda dar una buena presentación a tu página.



9. Video

En HTML se introduce la interesante posibilidad de mostrar videos directamente desde nuestro navegador. De hecho, si arrastramos un video a la ventana del navegador, veremos que comienza a reproducirse en él. Para poder insertar videos en nuestras páginas HTML tenemos que utilizar la etiqueta <video> y de la misma manera que <audio> e en la etiqueta <video> tendremos que utilizar el atributo “src”.

Entonces siguiendo la misma estructura de <audio> el resultado será el siguiente:

```
<video src = "nombre del archivo.mp4" controls />
```

```
<video src = "nombre del archivo.mp4" autoplay />
```

Esos son unos ejemplos de cómo podemos construir la etiqueta video, pero de la misma manera que audio contamos con los atributos:

Autoplay: Se reproduce de manera automática al cargar la página.

Control: Permite mostrar los controles básicos del video como son, play/pause, volumen y barra de carga.

Loop: Al terminar el video este se vuelve a reproducir en bucle.

Mute: Se silencia de manera automática.

Los formatos soportados por los navegadores suelen ser:

Formato	Codec utilizado	Características	¿Recomendado?
MP4	x264, DivX H264	Alta calidad. Codec x264 libre.	Sí, buen soporte
WebM	VP8, VP9	Alternativa libre a MP4 de Google.	Sí, soporte medio
Ogv	Theora	Alternativa libre a MP4.	Sí, soporte medio
MKV	Matroska	Buena compresión. Potente.	No, alto consumo CPU
AVI	XviD, DivX 3/5	Menor compresión que MP4.	No
HEVC	x265, DivX HEVC	Futura evolución de MP4.	No, en desarrollo aún.

Práctica 1: Video con letra. Crea un nuevo archivo HTML y en este colocaras el video de una canción y en tu página colocaras la letra.



Práctica 2: Película. Crea un nuevo archivo HTML y en este colocaras 1 el tráiler de una película y escribirás la reseña, recuerda dar buena presentación a tu página.



10. Formularios

El formulario es una de la parte más complicada de HTML y es una herramienta muy buena por lo mismo.

Esta información se obtiene mediante ayuda de: botones, cajas de texto, casillas de verificación, botones radio, etc.

Los formularios son una forma de obtener datos de parte del usuario para después poder procesarlo en algún servidor, muchas páginas cuentan con uno, sobre todo si es a la hora de registrarte en alguna, páginas como Facebook, Google, YouTube, Twitter, páginas de gobierno, páginas de escuelas y la lista continua.

Ejemplo de Facebook:

Registrarte

×

Es rápido y fácil.

Fecha de nacimiento ?

Sexo ?

☐ Mujer

☐ Hombre

☐ Personalizado

Es posible que las personas que usan nuestro servicio hayan subido tu información de contacto a Facebook. [Obtén más información.](#)

Al hacer clic en "Registrarte", aceptas nuestras [Condiciones](#), la [Política de privacidad](#) y la [Política de cookies](#). Es posible que te enviemos notificaciones por SMS, que puedes desactivar cuando quieras.

Registrarte

Iniciar sesión

[¿Olvidaste tu contraseña?](#)

Crear cuenta nueva

10.1 Form

La etiqueta que delimita un formulario es la etiqueta **<form>** con su respectivo cierre **</form>**. La etiqueta **<form>** es un elemento de bloque. En su interior puede haber cualquier elemento típico de una página web (párrafos, imágenes, divisiones, listas, tablas, etc.), además de las etiquetas que crean los controles.

Las etiquetas que crean los controles en los formularios son **<input />** (no necesita cierre), **<button>**, **<select>**, **<optgroup>**, **<option>** y **<textarea>**. Por último, la etiqueta **<label>** permite mejorar la accesibilidad de los controles.

10.1.1 Input **<input />**

Es la etiqueta con la cual podremos realizar varias de las funciones más importantes de los formularios, como cajas de texto, passwords, botones, botones radio, colores, etc. No necesita cierre y va dentro del bloque de **<form>**

10.1.2 Atributos comunes del Formulario

Hay ciertos atributos a conocer de los formularios, ya sea porque es obligatorio o por mejorar las funciones de los formularios hay que recordar cuales son y para que funcionan.

10.1.2.1 El atributo “type”

El atributo type de la etiqueta **<input />** indica el tipo de control de que se trata, si es: botón para enviar “submit”, botón para reiniciar “reset”, texto “text”, contraseña “password”, archivo “file”, imagen “image” y oculto “hidden”, botón “button”, número “number”, búsqueda “search”, teléfono “tel”, dirección web “url”, dirección de correo “email”, fecha “date”, hora “time”, color “color” y rango

“range”, mes “month”, semana “week” y fecha y hora local “datetime-local”.

10.1.2.2 El atributo name

El atributo name identifica al control. Todos los controles de un formulario deben tener el atributo name porque, en general, sólo se envían los controles que lo tienen (aunque hay excepciones, como los controles de tipo imagen o los botones, que no los necesitan).

En general, los atributos name no se deben repetir, porque no se recibiría toda la información al enviarla. La única excepción es el botón radio, como se explica en el apartado correspondiente.

10.1.2.3 El atributo value

El atributo value permite establecer el valor inicial de un control, aunque cada control lo utiliza de una forma ligeramente distinta y algunos controles no lo admiten (área de texto, archivo o imagen).

En algunos casos (botones) el valor se muestra al usuario y el usuario no puede modificarlo. En otros casos (cajas de texto, contraseña) el valor se muestra en la página y el usuario puede modificarlo. En otros casos (casillas de verificación, botones radio, opciones de menú, oculto) el valor no se muestra en la página y el usuario no puede modificarlo.

En algunos casos (botones, oculto, botones radio) el atributo value es necesario. En otros casos (casilla de verificación, opciones de menú) es conveniente. En otros casos (cajas de texto, contraseña, imagen) puede omitirse. En el caso del selector de archivo, los navegadores no lo admiten por motivos de seguridad.

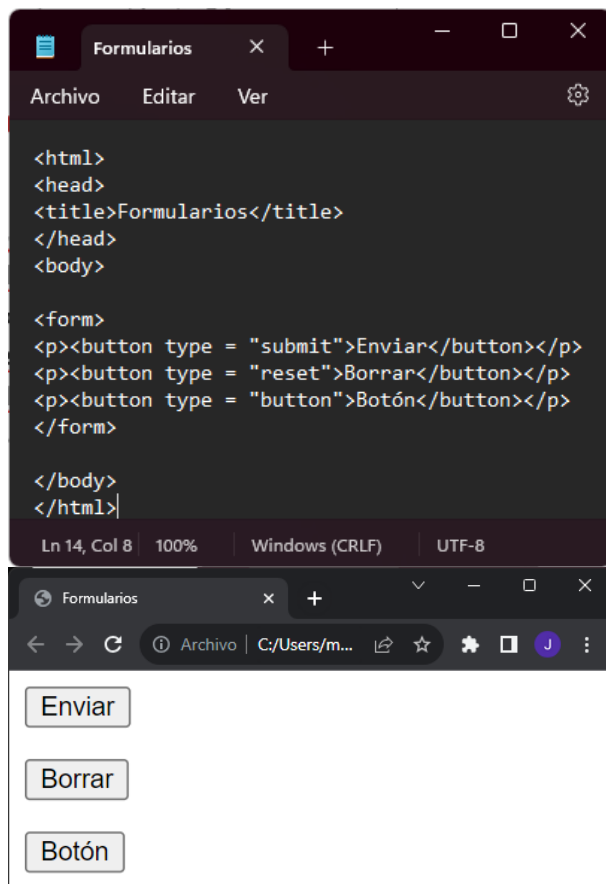
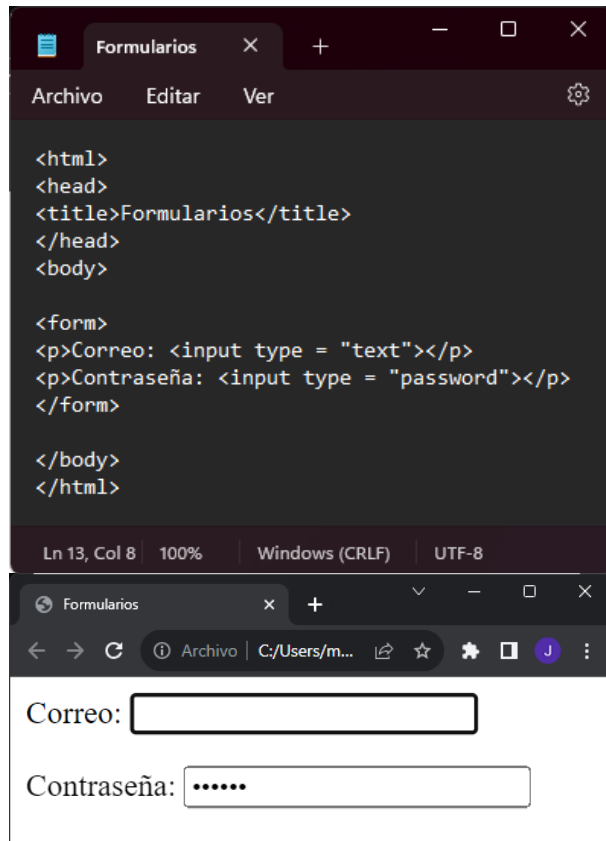
10.2 Cajas de texto

Ya viendo la parte más compleja de la teoría será bueno empezar con la práctica y nada mejor para iniciar que las cajas de texto. En estas cajas el usuario puede introducir datos textuales de gran importancia.

Como vimos esta se realiza dentro del bloque `<form>` con la etiqueta `<input>` y el atributo `"type"` tendrá el valor de `"text"`.

También tenemos el cuadro de texto y es tipo contraseña, este se diferencia por ocultar los caracteres que introducimos.

Esta se hace de la misma manera que el cuadro normal pero el valor de `"type"` será `"password"`.



10.3 Botones

La función de botones se incorpora mediante la etiqueta `<button>` esta etiqueta permite crear botones de tipo **submit** o **reset** o botones de tipo general que deben asociarse a scripts para hacer algo. Los botones **submit** o **reset** se crean mediante el atributo `type` con el valor **submit** o **reset**. El botón de uso general se crea mediante el atributo `type` con el valor **button**.

Práctica 1: Formulario 1. Crea un nuevo archivo HTML y transcribe este formulario, recuerda que la contraseña deben ser caracteres ocultos.



Ingresar los datos

Nombres

Apellidos

Dirección

Correo

Teléfono

Contraseña

Repita la contraseña

10.4 Botones radio

Cuando quieras mostrar al visitante una lista de valores de entre los que tiene que elegir uno solo, tienes que usar botones de opción. Dentro del código HTML que define este botón de opción tendrás que indicar cada uno de esos valores. La mayoría de los formularios utilizan estos pues facilita llevar un control de datos al solo mantener un pequeño número de opciones como digamos, el género.

Para utilizar los botones de radio lo haremos mediante la etiqueta `<input>` con el atributo `“type”` con el valor de `“radio”`, por primera vez de forma obligatoria tendremos que agregar el atributo `“name”` y se recomienda agregar el atributo `“value”` pues ya que las opciones que agreguemos tendrán un mismo `“name”` necesitamos que tengan un valor diferente.

Fíjate en el siguiente ejemplo y replícalo:

Formularios

Archivo Editar Ver

```
<html>
<head>
<title>Formularios</title>
</head>
<body>

<p>Ocupación</p>

<form>
<p><input type="radio" name="ocupacion" value="1">Estudio</p>
<p><input type="radio" name="ocupacion" value="2">Trabajo</p>
<p><input type="radio" name="ocupacion" value="3">Estudio y trabajo</p>
<p><input type="radio" name="ocupacion" value="4">Ninguno</p>
</form>

</body>
</html>
```

Ln 17, Col 8 | 100% | Windows (CRLF) | UTF-8

Formularios

← → ↻ ⓘ Archivo | C:/Use

Ocupación

☐ Estudio

☐ Trabajo

☐ Estudio y trabajo

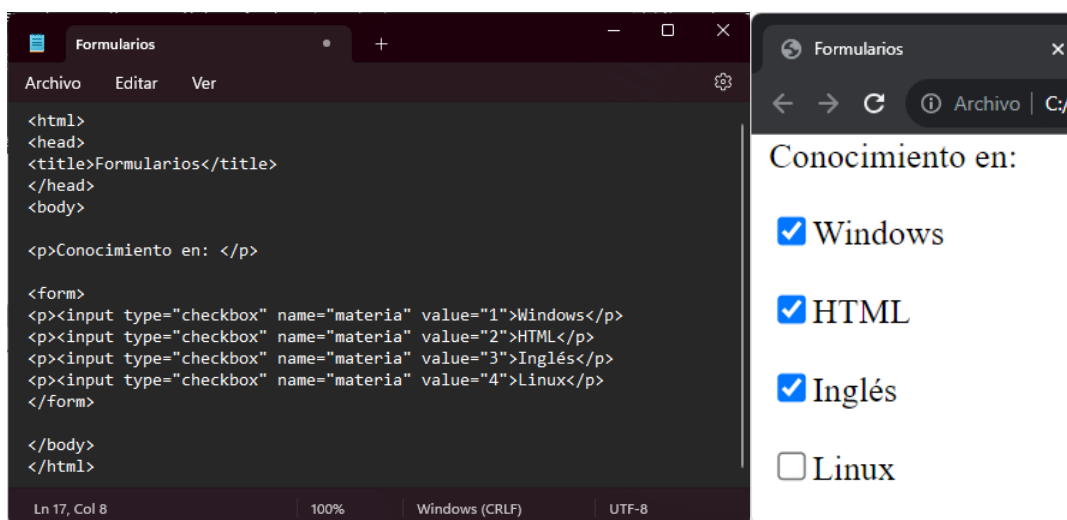
☐ Ninguno

10.5 Casillas de verificación

Las casillas de verificación son muy parecidas a los botones radio pero con la gran diferencia de que en estas SI se puede elegir más de una opción, eso y las casillas de verificación son cuadradas y no circulares.

Estas se ingresan con la etiqueta `<input>` y con el atributo `"type"` con un valor igual a `"checkbox"`, se aconseja utilizar el atributo `"name"` para proporcionar una identidad a la casilla, El atributo `"value"` contiene el valor que envía el formulario si la casilla de verificación está marcada. Si el atributo `"value"` no está establecido, el formulario envía el valor `"on"`.

Observa el siguiente ejemplo y hazlo tú también:



Práctica 2: Formulario 2. Crea un nuevo archivo HTML y realiza el código para recrear el siguiente formulario, usando los distintos tipos de `<input>` y `<button>`.



Rellene por favor el siguiente formulario con atención y sin dejarse ningún dato

Name: contraseña:

Estado civil:

☐ soltero ☐ casado ☐ separado ☐ viudo

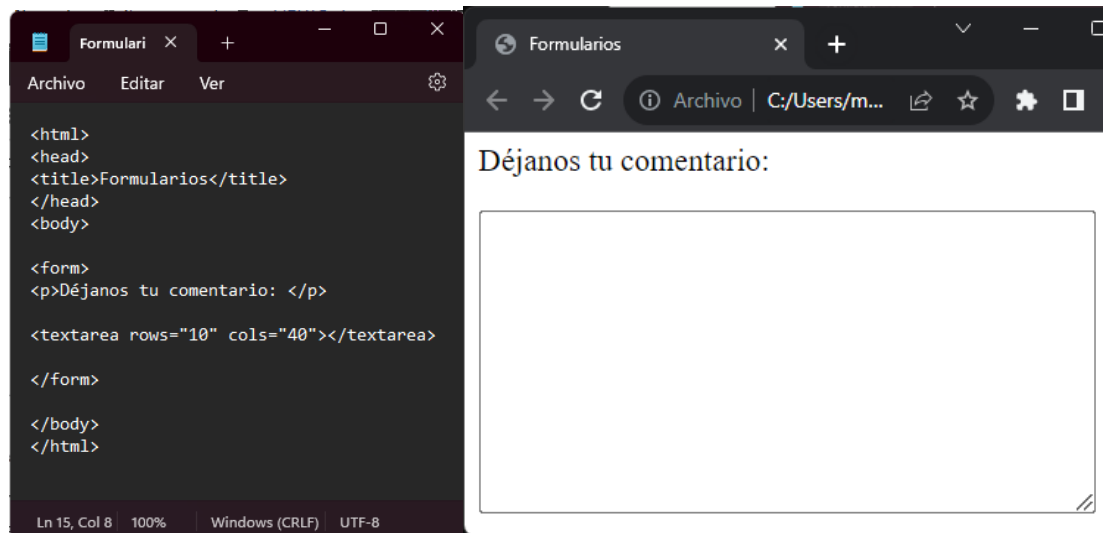
Número de hijos:

☐ ninguno ☐ uno ☐ dos ☐ mas de 2

10.6 Área de texto

Las cajas de texto es un espacio en donde se puede introducir información de manera más extensa pues como su nombre lo dice el área de texto puede ser tan grande como lo requieras, las cajas de texto se componen de varias líneas se crean mediante la etiqueta `<textarea>` en este caso la etiqueta de cierre `</textatera>` es obligatoria. Los atributos obligatorios “rows” (filas) y “cols” (columnas) establecen el número de filas y columnas iniciales de la caja, aunque los navegadores permiten modificarlo arrastrando la esquina inferior derecha.

Aquí un ejemplo:



Si se escribe texto en el interior de la etiqueta, este se muestra en la caja de texto. HTML 5 introdujo el atributo “**placeholder**” que puede ser más cómodo para el usuario si tiene que borrar todo el texto incluido.

Práctica 3: Formulario 3. Crea un nuevo archivo HTML, transcribe este texto dentro del bloque `<body>` y escribe en tus apuntes el código fuente de tu archivo ,especifica cual es la diferencia entre Textarea 1 y Textarea 2 y escribe si se puede tener texto prescrito dentro de las cajas `<input type =”text”>`



`<form>`

`<textarea rows="5" cols ="40" >Textarea 1</textarea>`

`<textarea rows="5" cols ="40" placeholder = "Textarea 2 "></textarea>`

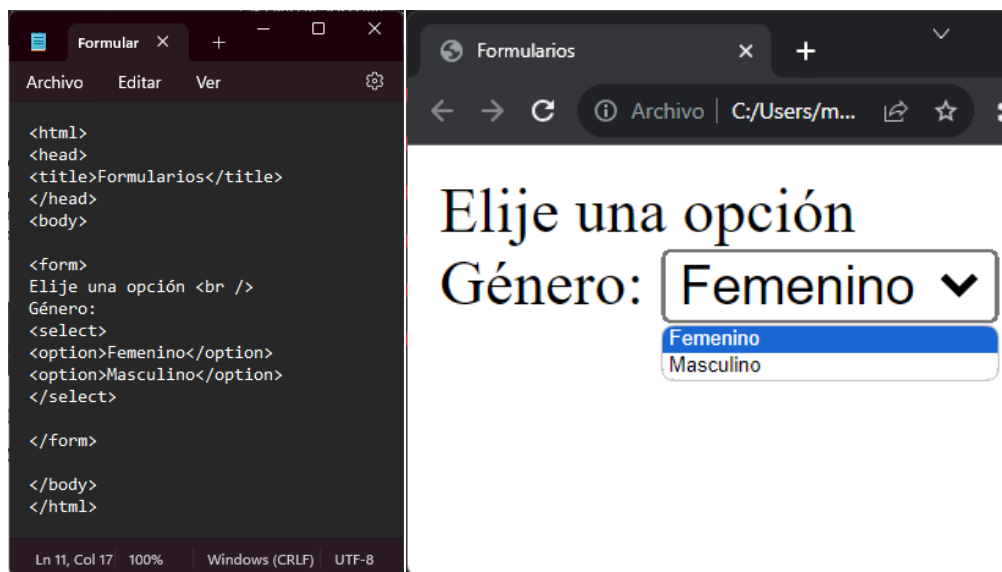
`</form>`

10.7 Menú de selección

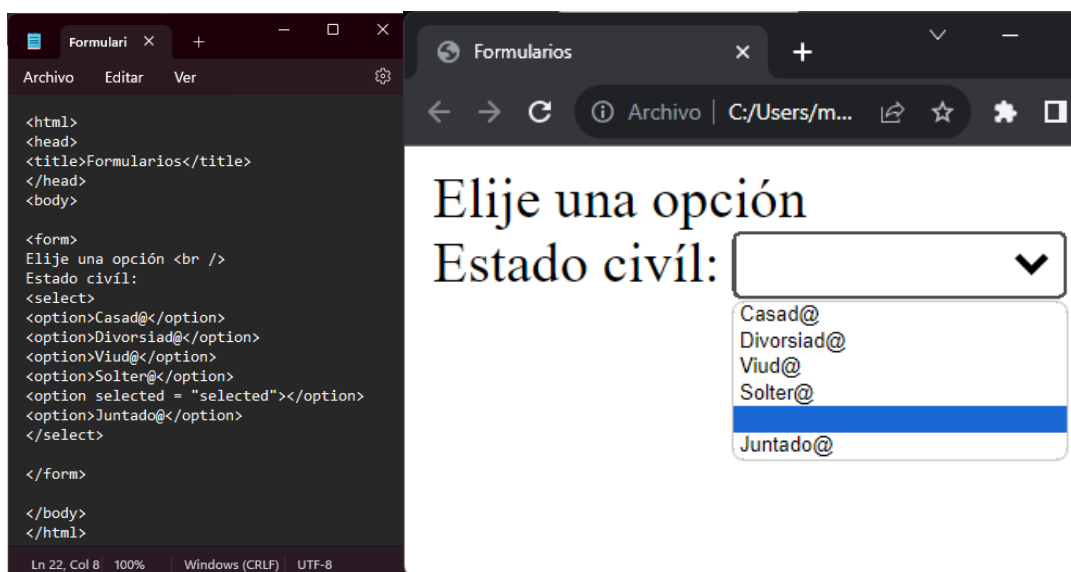
El menú de selección o listas desplegables son elementos de selección agrupados en un cuadro (o lista) en que se presentan diferentes opciones a elegir de este se pueden elegir 1 o más si está activado el atributo “**multiple**”.

Para crear un menú de selección hay que conocer 2 etiquetas muy importantes, la primera es **<select>** esta es la etiqueta con la que abriremos el menú, dentro de esta etiqueta introduciremos las opciones a elegir con la etiqueta **<option>**, ambos con su respectivo cierre.

Aquí el ejemplo, realiza el ejemplo:

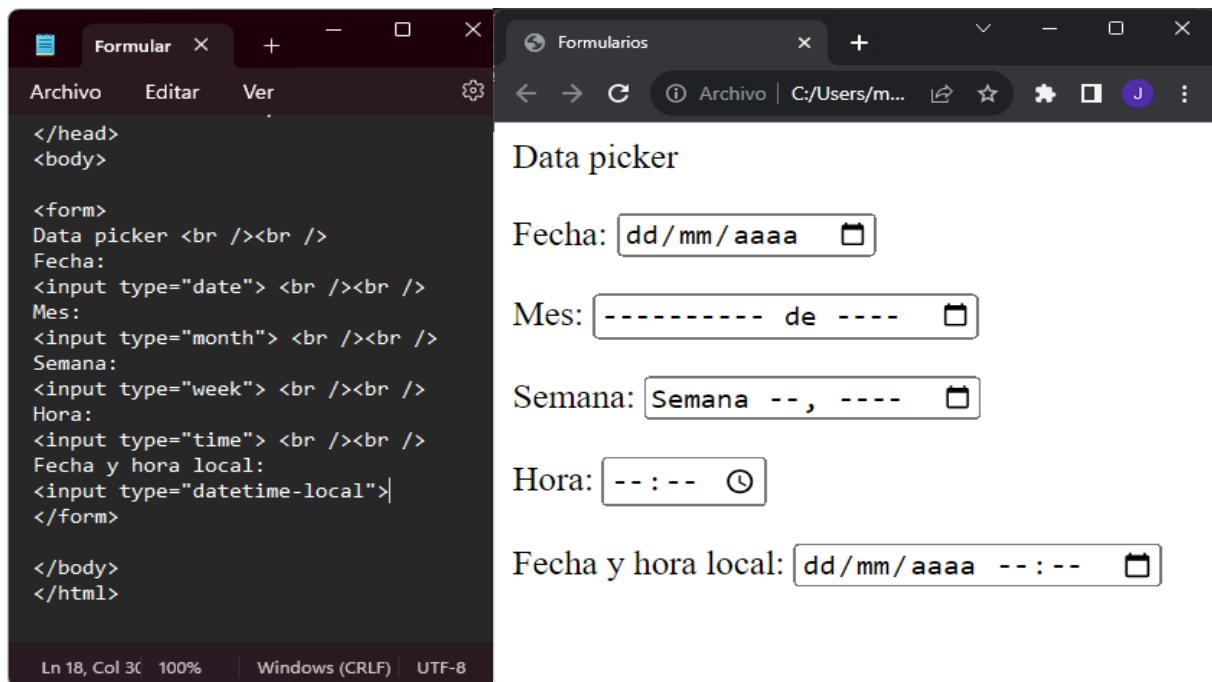


Podemos fijar una la opción que queramos para que el menú aparezca con dicha opción gracias al atributo “**selected = selected**”, aunque el problema de esto es que no sabremos si el usuario eligió la opción o se eligió por default, por eso es mejor dejar una opción en blanco al principio. Observa el ejemplo y créalo:



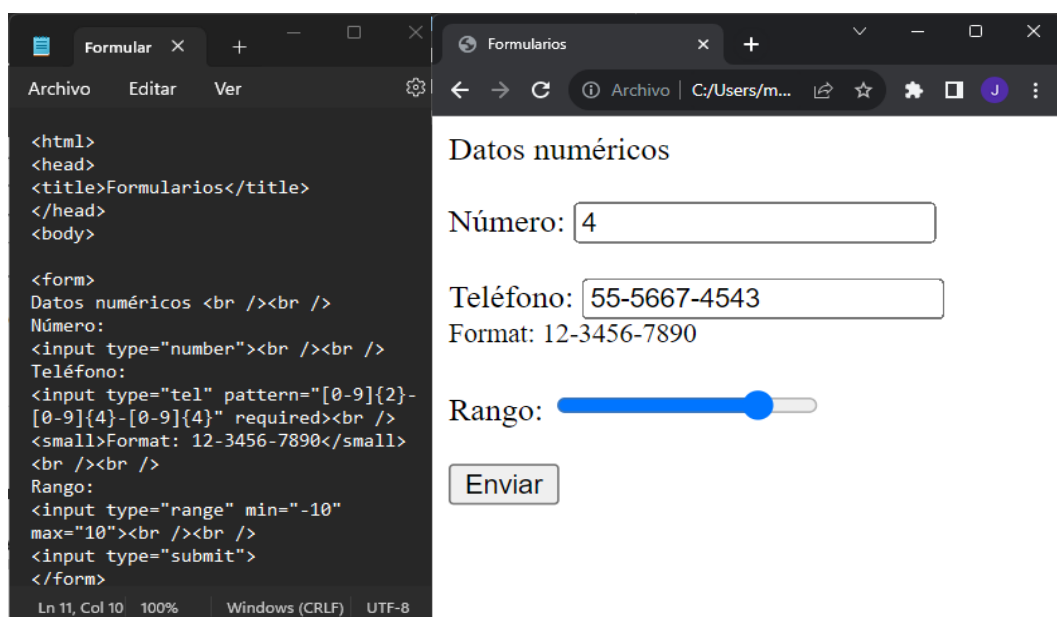
10.8 Fecha y hora

Para que usuario final pueda ingresar una fecha, bien podría utilizarse simplemente una etiqueta `<input>` y el atributo `"type"` con el valor `"text"`, pero el usuario podría introducir cualquier dato que no fuera una fecha, por ello existen distintos tipos de `"type"` para la fecha. A continuación, un ejemplo con los tipos de selector de fecha, denominado frecuentemente **date picker** (realiza el ejemplo).



10.9 Números

Para poder ingresar datos numéricos existen distintas formas dependiendo del uso que se le quiera dar, por ejemplo, si lo que se desea ingresar es un número telefónico, se puede emplear una etiqueta `<input>` y el atributo `"type"` con el valor `"tel"`. A continuación, un ejemplo por los tipos de entrada numérica (elabora el ejemplo):



10.10 Validación de datos para formularios

Como hemos estado aprendiendo, todos estos tipos de datos dentro de la etiqueta **<form>** sirven para crear justamente un formulario, por ello es que la validación de formularios es una parte crucial en el desarrollo web para asegurarse de que los datos ingresados por los usuarios sean correctos y cumplan con ciertos criterios.

Existen diversos atributos para la etiqueta **<input>**, como lo son **"pattern"**, **"min"**, **"max"** y **"required"**. Sus usos son los siguientes:

Atributo	Utilidad
pattern (Patrón)	Este atributo indica expresión regular que debe cumplir lo que se ingrese el usuario, es decir, un patrón que debe cumplir, lo cual evitará que usuario cometa algún error al ingresar sus datos.
min (Mínimo) y max (Máximo)	Este atributo nos indica en los datos numéricos el rango entre el valor mínimo y máximo que se puede ingresar.
required (Requerido)	Este atributo nos indica que es obligatorio el ingresar determinado dato, ya que al presionar el botón de enviar (submit), lanzará una alerta de que el campo está vacío.

A continuación, un ejemplo del uso de las validaciones (recrea el código que se presenta). Observa la etiqueta **<hr />**, la cual puede hacer una línea de separación:

Formularios

Archivo Editar Ver

```
<html>
<head>
<title>Formularios</title>
</head>
<body>

<form>
<h3>Validación de datos</h3>
<hr />
Ingrese su correo electrónico:
<input type="email" name="email" required
pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$">
<br /><br />
Ingrese su edad:
<input type="number" name="age" required min="18"
max="99"><br /><br />
<input type="submit" value="Enviar">
</form>

</body>
</html>
```

Ln 17, Col 8 100% Windows (CRLF) UTF-8

Formularios

← → ↻ ⓘ Archivo C:/Users/m...

Validación de datos

Ingrese su correo electrónico:

Ingrese su edad:

!

Incluye un signo "@" en la dirección de correo electrónico. La dirección "m" no incluye el signo "@".

10.10.1 Expresiones regulares para <input> (pattern)

Las expresiones regulares son patrones de texto que se utilizan para buscar y validar cadenas de caracteres según ciertos criterios. En el contexto del atributo pattern, estas expresiones regulares definen el formato que debe tener el valor ingresado en un campo de formulario para considerarse válido.

Se presenta una descripción general de algunos de los elementos y más comunes que puedes usar en las expresiones regulares:

Caracteres Literales: Los caracteres que no son meta caracteres representan ellos mismos. Por ejemplo, la expresión **abc** buscará la secuencia "abc" en el texto.

Clases de Caracteres: Se definen entre corchetes **[]** y representan un conjunto de caracteres posibles. Por ejemplo, **[aeiou]** coincide con cualquier vocal.

Rangos de Caracteres: Puedes definir rangos de caracteres utilizando el guion - dentro de una clase de caracteres. Por ejemplo, **[0-9]** coincide con cualquier dígito.

Meta caracteres:

.: Coincide con cualquier carácter excepto nueva línea.

*****: Coincide con cero o más repeticiones del elemento anterior.

+: Coincide con una o más repeticiones del elemento anterior.

?: Coincide con cero o una repetición del elemento anterior.

^: Coincide con el comienzo de la línea.

\$: Coincide con el final de la línea.

****: Se utiliza para escapar meta caracteres y caracteres especiales.

Agrupación: Los paréntesis **()** se utilizan para agrupar elementos y aplicar operadores a grupos de caracteres. Por ejemplo, **(abc)+** coincide con una o más repeticiones de "abc".

Operador OR: Se utiliza el símbolo **|** para indicar alternativas. Por ejemplo, **(perro|gato)** coincide con "perro" o "gato".

Cuantificadores: Se utilizan para especificar cuántas repeticiones de un elemento se deben encontrar.

{n}: Coincide exactamente con n repeticiones del elemento anterior.

{n,}: Coincide con al menos n repeticiones del elemento anterior.

{n,m}: Coincide con al menos n y máximo m repeticiones del elemento anterior.

Práctica 4: Formulario 4. Crea un nuevo archivo HTML, estructura el código para que recrear el siguiente formulario. Recuerda utilizar validaciones.



Formulario de contrato	
Nombre	Solo su(s) nombre(s)
Apellido	Paterno
Apellido	Materno
Género	
<input type="radio"/> Masculino <input type="radio"/> Femenino	
Fecha de nacimiento	Día: <input type="text"/>
	Mes: <input type="text" value="Enero"/>
	Año: <input type="text"/>
Dirección	
<input type="text"/>	
Correo electrónico	
<input type="text"/>	
Cualidades	
Ej. Honesto, solidario, leal. <input type="text"/>	
Teléfono	<input type="text"/>
Escolaridad	<input type="text" value="Primaria"/>
Fecha de egreso	<input type="text" value="dd/mm/aaaa"/>
¿Qué tan comprometido está en cumplir su trabajo?	<input type="range" value="1"/> 1 <input type="range" value="5"/> 5
Contraseña	<input type="text"/>
Repite la contraseña	<input type="text"/>

☐ Acepto los términos y condiciones

☐ Quiero recibir mensajes sobre la empresa

11. Práctica final

Crea un nuevo archivo HTML, con la temática que tú quieras utilizando todo lo que has aprendido a lo largo de este manual, recuerda que se evaluará la habilidad de utilizar tus conocimientos y tu creatividad.



¡¡Suerte!!

12. Evaluación

HTML (HyperText Markup Language)



HOJA DE EVALUACIÓN			
Alumno:		Día(s):	
Prof.(a):		Horario:	
Fecha	Tema	Calificación	Firma Prof.(a)
	Textos En HTML <p>, <pre>		
	Títulos <h1>,<h2>...		
	Fuente 		
	Imágenes 		
	Marquesinas <Marquee>		
	Listas ,,<dl>		
	Tablas <table>		
	Enlaces <A>		
	Audio <audio>		
	Video <video>		
	Formulario <form>		
	Practica final		
	Calificación Examen Semestral:		
	Promedio:		