

RESUMEN HTML

Por Javier Peiro Aguado (2DAW)

1- ¿Qué es HTML?

Es el lenguaje básico que se utiliza para construir las páginas web que vemos en internet. Imagina que es como un conjunto de instrucciones que le damos a un navegador (como Chrome, Firefox o Safari) para que muestre el contenido de una manera organizada.

2-Estructura de una página HTML:

- HTML: Lenguaje para escritura de páginas Web.
- Estructura básica: html, head, title, meta, body.
- Etiquetas: Entornos que rodean el contenido y envían instrucciones al navegador.
- Head (< head >): Contiene datos de páginas (título, codificación).
- Body (< body >): Muestra el contenido de página que usted puede ver.
- Mayúsculas y minúsculas: HTML no es sensible a mayúsculas y minúsculas.
- DOCTYPE – Indica al navegador que el documento es un HTML.
- Charset — Define el código de caracteres. El más común es UTF-8.

3-Salto de línea

 en HTML crea un salto de línea.

Sirve para separar texto en diferentes líneas dentro de una página web.

Se escribe
 donde quieras que empiece una nueva línea.

Es útil para formatear textos, direcciones, poemas y otros contenidos donde la separación en líneas es importante.

Ejemplo:

HTML

<body>

Esto es una línea.

Esto es otra línea.

</body>

4-Párrafo<p>

El elemento <p> en HTML define párrafos, que son bloques de texto separados por un espacio respecto a otros párrafos. Dentro de un párrafo se pueden incluir saltos de línea con
.

Ejemplo:

<p>

Prrafo de ejemplo 1

</p>

<p>

parrafo de ejemplo 2

</p>

- <p> proviene de *paragraph*.
- No es recomendable usar únicamente
 para dar formato; es mejor estructurar con párrafos y añadir saltos dentro de ellos si es necesario.

5-Títulos h1, h2, h3, h4, h5, h6

Los elementos <h1> a <h6> en HTML se usan para definir títulos, siendo <h1> el de mayor nivel y tamaño y h6 el menor. El navegador determina el tamaño de la fuente según el nivel del título.

- Cada título se muestra en una línea nueva, sin importar cómo se escriba en el código.
- Los motores de búsqueda usan los títulos para identificar los temas de la página.
- <h1> significa *heading*, que significa título.
- No son los títulos del contenido con el título de la página, que se define en <head>.

6-Énfasis ()

Los elementos y en HTML se usan para enfatizar texto:

- aplica un énfasis más fuerte (usualmente en **negrita**).
- aplica un énfasis menos fuerte (generalmente en *itálica*).

Características:

- No generan saltos de línea como los títulos.
- proviene de *emphasis* (énfasis) y significa fuerte.

7 - Hipervínculo a otra página del mismo sitio <a>

El elemento más importante en una página web es el hipervínculo (<a>), que permite navegar entre páginas. Esto diferencia a una página web de un libro, ya que en un sitio web podemos tener distintas alternativas de recorrido.

Ejemplo:

```
<a href="pagina2.html">Noticias</a>
```

- El texto entre las etiquetas aparece subrayado por defecto, y al hacer clic, se carga la página indicada.
- Introduce el concepto de propiedad, como href, que tiene un nombre y un valor (en este caso, la URL), el cual va entre comillas.

8-Hipervínculo a otro sitio de internet <a>

Para crear un hipervínculo que dirija a otro sitio web, la sintaxis es:

```
<a href="http://www.google.com">Buscador Google</a>
```

- La propiedad href incluye el **protocolo** (como http) seguido del dominio del sitio (www.google.com).
- Si no se especifica una página dentro del dominio, el servidor devuelve la página principal configurada por defecto.
- También se pueden enlazar páginas específicas de un dominio, indicando la ruta exacta. Por ejemplo:

html

Copiar código

```
<a href="http://tutorialesprogramacionya.com/acercade.html">Acerca de Google</a>
```

Hay que conocer el nombre de la página y, si está en una subcarpeta, proporcionar la ruta completa.

9- Imágenes dentro de una página

Para insertar imágenes en una página web se utiliza la etiqueta , que no necesita cierre (igual que
). Esta etiqueta tiene dos propiedades básicas:

- **src:** Indica la ruta del archivo de imagen. Si la imagen está en el mismo directorio que la página, basta con poner el nombre del archivo. En sistemas Linux, recuerda que los nombres de archivos son sensibles a mayúsculas y minúsculas, así que usa minúsculas para evitar problemas.
- **alt:** Proporciona una descripción de la imagen. Es útil para navegadores que no muestran imágenes o usuarios con accesibilidad especial.

Ejemplo básico:

```

```

- `` viene de *image* (imagen).
- `src` significa *source* (fuente).
- `alt` es de *alternative* (alternativa).

10-Hipervínculo mediante una imagen `<a>` y ``

Podemos combinarlos para crear enlaces donde, en lugar de texto, se muestre la imagen.

Esto se logra colocando la etiqueta `` dentro de las etiquetas `<a>`.

Ej:

```
<a href="pagina1.html"></a>
```

11-Apertura de un hipervínculo en otra pestaña del navegador

La propiedad `target` del elemento `<a>` permite especificar dónde se abrirá el enlace. Para abrirlo en una nueva pestaña, usamos el valor `"_blank"`.

Ej:

```
<a href="pagina2.html" target="_blank">Abrirá esto en otra pestaña</a>
```

12-Hipervínculo a un cliente de correo `<a>`

La etiqueta <a> también puede usarse para abrir un programa de correo configurado en la computadora.

Ej:

```
<a href="http://www.clarin.com.ar" target="_blank">Periódico Clarín</a>
```

13-Anclas llamadas desde la misma página

En HTML, podemos crear anclas dentro de una página para permitir que los usuarios se desplacen fácilmente entre diferentes secciones, especialmente en páginas largas.

Ej:

```
<a name="nombreancla"></a> (Sería el sitio al que queremos ir tras pulsar el vínculo)
```

```
<a href="#nombreancla">Texto del enlace</a>(Sería el vínculo)
```

14-Anclas llamadas desde otra página:

Es posible enlazar anclas desde otras páginas. Para ello, simplemente agregamos el nombre de la página y el ancla:

```
<a name="introduccion"></a>
```

```
<a href="pagina2.html#introduccion">Introducción</a>(En la otra página)
```

15-Lista Ordenada ()

En HTML podemos crear **listas ordenadas** con la etiqueta , que numera los ítems automáticamente:

```
<ol>
```

```
  <li>Primer ítem</li>
```

```
  <li>Segundo ítem</li>
```

```
</ol>
```

16-Lista NO Ordenada ()

En HTML podemos crear **listas no ordenadas** con la etiqueta , que usa símbolos en lugar de números:

```
<ul>
  <li>Primer ítem</li>
  <li>Segundo ítem</li>
</ul>
```

17-Lista de definiciones(<dl>)

La lista de definiciones (<dl>) se usa para asociar un término con su definición. El navegador destaca y separa ambos elementos. Se utiliza comúnmente para crear definiciones, como la de varios lenguajes de programación.

- <dl> viene de definition list
- <dt> viene de definition term
- <dd> viene de definition description

18-Listas anidadas

HTML deja anidar listas dentro de otras, como por ejemplo tener una lista ordenada con elementos que incluyen listas no ordenadas.

Ej:

```
<li>Diarios
  <ul>
    <li><a href="http://www.lanacion.com.ar">La Nación</a></li>
    <li><a href="http://www.clarin.com.ar">Clarín</a></li>
  </ul>
</li>
```

Poniendo indentación o sangrado (espacios en blanco a la izquierda) mejorará la legibilidad del código, pero el navegador no lo toma en cuenta. Esto ayuda a identificar fácilmente el inicio y fin de cada marca HTML.

19- Tabla(<table><tr><td>)

Las tablas en HTML se utilizan para organizar datos en filas y columnas. No deben usarse para el diseño completo de una página, ya que eso se hace con CSS. Los elementos clave para crear una tabla son:

- <table>: Marca de inicio de la tabla, que debe cerrarse.
- <tr>: Define una fila dentro de la tabla, también necesita cerrarse, viene de table row.
- <td>: Es una celda dentro de una fila, que también debe cerrarse, viene de table data.

20-Tabla con encabezado(<th>)

En una tabla, las celdas de encabezado se indican con el elemento <th>, en lugar de <td>, para marcar los títulos de cada columna. Estas celdas se encuentran dentro de una fila, es decir, entre las marcas <tr> y </tr>. Los navegadores muestran las celdas de encabezado de manera diferente a las de los datos.

- <th> proviene de "table header cell" (celda de encabezado de tabla).

21-Tabla con título(<caption>)

Para agregar un título a una tabla, utilizamos el elemento <caption>, que debe colocarse justo después de abrir la etiqueta <table>. Este elemento requiere tanto la marca de apertura como la de cierre.

Por ejemplo, para una tabla que muestre la población de varios países, se usaría:

<caption>Población de distintos países</caption>

- <caption> significa título.

22-Tabla y combinación de celdas

En algunos casos, es necesario que una celda ocupe el espacio de varias celdas, ya sea de manera horizontal o vertical. Para esto, se utilizamos las propiedades `colspan` y `rowspan` en las etiquetas `<td>` o `<th>` y le asignamos un valor entero, generalmente a partir de 2.

- **colspan** se usa para que una celda ocupe varias columnas. Por ejemplo, si queremos que una celda ocupe tres columnas, usamos:

```
<td colspan="3">Facturación de los últimos tres meses</td>
```

- **rowspan** se usa para que una celda ocupe varias filas. Por ejemplo, para que una celda ocupe tres filas:

```
<td rowspan="3">Secciones</td>
```

Esto permite combinar celdas en tablas para una presentación más organizada y eficiente.

23- Contenido de la cabecera de la página(<title>)

El elemento `<title>` se utiliza para definir el título que aparecerá en la barra del navegador, en la parte superior de la ventana. Aunque la cabecera del archivo HTML contiene información que generalmente no se muestra en la página, el título es una excepción, ya que es visible y ayuda a identificar de qué trata la página.

Es importante elegir un título claro y relevante para el contenido que se presenta. Aquí tienes un ejemplo de una página simple con un título y un hipervínculo a otra página con un título diferente:

24- Contenido de la cabecera de la página(<meta>)

El elemento `<meta>` se utiliza en la cabecera de una página HTML para proporcionar metadatos, como la codificación de caracteres y otra información relevante para los motores de búsqueda y los navegadores. Un uso común es definir la codificación de caracteres de la página:

<meta charset="UTF-8">

Hay muchos metas pero hay algunos más comunes como author, copyright o description.

25-Comentarios<!-- -->

Los comentarios en HTML son textos que los navegadores ignoran y están pensados para facilitar el trabajo del desarrollador. Son útiles para dejar anotaciones, explicar partes del código o deshabilitar temporalmente fragmentos durante el desarrollo.

Usos habituales de los comentarios:

1. **Notas explicativas:** Ayudan a documentar el código.
2. **Deshabilitar código temporalmente:** Permiten probar cambios sin eliminar partes del código.

<!-- Ejemplo de comentario -->

26-Sintaxis para caracteres especiales

En HTML, no podemos usar directamente ciertos caracteres especiales, como < y >, porque el navegador los interpretaría como marcas HTML. Para solucionarlo, usamos entidades HTML, que son combinaciones de caracteres que representan símbolos específicos.

Principales caracteres y la sintaxis:

- < se escribe como <
- > se escribe como >
- & se escribe como &
- Espacio en blanco:
- Símbolo de copyright: ©
- Euro: €

27- Formulario-(<form>)

Un formulario permite a los visitantes de un sitio ingresar datos y enviarlos al servidor para su procesamiento, aunque HTML solo se encarga de construir el formulario. El registro y manejo de los datos se realiza con lenguajes de servidor como por ejemplo PHP, ASP, NodeJS, etc.

Elementos clave de un formulario:

1. Estructura básica

El elemento `<form>` define el formulario y utiliza las propiedades:

- **action:** Especifica la página que procesará los datos en el servidor (por ejemplo, `registrardatos.php`).
- **method:** Define cómo se envían los datos:
 - `post`: Datos enviados en el cuerpo de la solicitud (sin límite de tamaño).
 - `get`: Datos enviados en la URL (limitado en tamaño).

```
<form action="registrardatos.php" method="post">
```

2. Cuadros de texto

Para crear un cuadro donde el usuario pueda ingresar texto, usamos el elemento `<input>` con el atributo `type="text"`.

- **name:** Nombre identificador del campo.
- **size:** Tamaño visual del cuadro (cantidad de caracteres mostrados).

Ingrese su nombre:

```
<input type="text" name="nombre" size="20">
```

3. Botón de envío

Para enviar los datos al servidor, se utiliza `<input>` con el atributo `type="submit"`.

- **value:** Texto que aparecerá en el botón.

```
<input type="submit" value="enviar">
```

4. Cierre del formulario

El formulario se cierra con `</form>`.

Ej:

```
<form action="formEjemp.php" method="post">
```

Ingresa el nombre:

```
<input type="text" name="nombre" size="20">  
<br>  
<input type="submit" value="enviar">  
</form>
```

28- Formulario- input type="text"/"password"

`<input type="text">` Crea un campo tipo texto que te permite introducir cadenas de texto dentro

Usa `<input type="password">` es igual que el tipo texto, pero para que los caracteres ingresados aparezcan como asteriscos.

- Ejemplo:

```
<input type="password" name="clave" size="12">
```

2. Botón de reinicio:

Usa `<input type="reset">` para borrar todos los datos del formulario.

- Ejemplo:

```
<input type="reset" value="borrar">
```

3. Atributo name:

Cada campo debe tener un nombre único para ser identificado en el servidor.

Formulario de ejemplo:

```
<form action="procesar.php" method="post">  
  Usuario: <input type="text" name="usuario">  
  Contraseña: <input type="password" name="clave">  
  <input type="submit" value="enviar">  
  <input type="reset" value="borrar">  
</form>
```

29-Formulario- textarea

El elemento <textarea> permite ingresar texto en varias líneas, ideal para comentarios largos.

Ej:

```
<textarea name="comentarios" rows="5" cols="60"></textarea>
```

Propiedades:

- **name:** Identifica el campo en el servidor.
- **rows y cols:** Determinan el número de filas y columnas visibles.

30- Formulario- input type="checkbox"

El elemento checkbox permite al usuario seleccionar o deseleccionar opciones. Es ideal para elegir múltiples valores independientes.

Ej:

```
<input type="checkbox" name="java">Java<br>
```

Detalles:

- **type="checkbox":** Define el elemento como casilla de verificación.
- **name:** Identifica el control para su procesamiento en el servidor.
- Texto adicional puede colocarse junto al elemento para describir la opción.

31- Formulario- input type="radio"

Los botones de selección (radio) se usan cuando se necesita elegir solo una opción dentro de un grupo de opciones.

Ej:

```
<input type="radio" name="estudios" value="1">Sin estudios<br>  
<input type="radio" name="estudios" value="2">Primario<br>  
<input type="radio" name="estudios" value="3">Secundario<br>
```

32- Formulario- select(cuadro de selección individual)

Con el elemento `<select>` crea un cuadro que permite elegir una opción entre varias.

Propiedades:

- **`<select>`:**

Se utiliza para abrir el cuadro de selección.

La propiedad `name` define el nombre que recibirá el servidor.

- **`<option>`:**

Define cada opción dentro del cuadro de selección.

La propiedad `value` indica el valor enviado al servidor al seleccionar esa opción.

El texto entre las etiquetas `<option>` es lo que se muestra en pantalla.

- **`size`:**

Por defecto, muestra una opción visible (tamaño 1).

Si se cambia el valor, muestra varias opciones simultáneamente, aunque solo permite seleccionar una a la vez.

33-Formulario-select (cuadro de selección múltiple)

El cuadro de selección múltiple permite que el usuario seleccione varias opciones de una lista, a diferencia de un cuadro de selección estándar, que solo permite una selección.

Ej:

```
<select name="trabajos[]" size="2" multiple>
```

El usuario puede seleccionar varias opciones manteniendo presionada la tecla `Ctrl` (en Windows) o `Cmd` (en Mac), y haciendo clic en opciones no consecutivas.

34- Formulario- select (agrupamiento de opciones)

El elemento `<optgroup>` permite agrupar varias opciones dentro de un cuadro de selección, lo cual es útil si este contiene numerosos elementos.

```
<select name="productos">
```

```
  <optgroup label="Aficiones">
```

```
    <option value="1">Warhammer</option>
```

```
    <option value="2">Juegos PC</option>
```

```
  </optgroup>
```

```
<optgroup label="Trabajos">
  <option value="4">Programador</option>
  <option value="5">Electricista</option>
</optgroup>
</select>
```

35- Formulario- button

El elemento `<button>` se utiliza para crear botones en un formulario. Ofrece más opciones que los elementos `<input type="submit">` o `<input type="reset">`, ya que permite incluir contenido dentro del botón, como texto, imágenes o elementos HTML más complejos.

36- Formulario-input type="button"

El elemento `<input type="button">` crea un botón de acción personalizada. A diferencia de los botones de tipo submit o reset, este botón no tiene ninguna función predefinida en HTML. Su comportamiento depende de un programa en JavaScript que le asigne una acción cuando el usuario haga clic en él. El atributo value especifica el texto que se muestra en el botón.

Ej:

```
<input type="button" value="Haz algo">
```

37- Formulario-input type="file"

El control de tipo file permite a los usuarios cargar archivos desde su dispositivo y enviarlos al servidor. Este control solo indica al navegador que debe permitir la selección de archivos, pero el servidor debe contar con un programa para recibir y almacenar el archivo.

Ej:

```
<input type="file" name="archi">
```

- **type="file"**: Define el campo para subir archivos
- **name**: Define el nombre del campo, que se utilizará para identificar el archivo en el servidor.

38- Formulario-input type="hidden"

El elemento `<input type="hidden">` crea un campo oculto que no es visible para el usuario en el formulario. Este tipo de campo se utiliza principalmente para almacenar datos que el servidor necesita recibir, pero que no deben ser modificados ni visualizados por el usuario.

- **name:** Define el nombre del campo oculto, utilizado para identificar el dato que se enviará al servidor.
- **value:** Almacena el valor que será enviado al servidor al enviar el formulario.

39- Formulario-Agrupamiento de controles

El elemento `<fieldset>` en HTML se utiliza para agrupar y recuadrar varios controles dentro de un formulario. Todos los controles que queremos agrupar deben ir entre las etiquetas `<fieldset>` y `</fieldset>`. Además, podemos añadir un título a este recuadro utilizando el elemento `<legend>`.

40- Formulario-agrupamiento de iniciales

En HTML, los controles de formularios pueden inicializarse con valores predeterminados. Para hacerlo:

- **Texto:** Se usa la propiedad `value` en un `<input type="text">`. Ejemplo: `<input type="text" value="aquí su nombre">`.
- **Textarea:** El texto inicial se coloca entre las etiquetas `<textarea>`. Ejemplo: `<textarea>Ingrese aquí su curriculum</textarea>`.
- **Checkbox:** Se marca el checkbox por defecto con la propiedad `checked`. Ejemplo: `<input type="checkbox" checked>`.
- **Radio:** Igual que el checkbox, pero solo un radio puede estar marcado por defecto. Ejemplo: `<input type="radio" checked>`.
- **Select (individual):** Se selecciona un valor por defecto en un `<option>` con la propiedad `selected`. Ejemplo: `<option value="2" selected>España</option>`.
- **Select (múltiple):** Varios elementos `option` pueden tener la propiedad `selected` para que estén seleccionados por defecto. Ejemplo: `<option value="1" selected>Rojo</option>`.
-

41- Formulario-orden de foco de controles

La propiedad `tabindex` permite definir el orden en que los controles de un formulario reciben el foco al presionar la tecla "Tab". Este valor, que puede ser un número entre 0 y 32767, establece el flujo de navegación entre los campos.

Por ejemplo:

- `<input type="text" name="text1" tabindex="1">`
- `<input type="text" name="text2" tabindex="4">`
- `<input type="text" name="text3" tabindex="7">`

Los valores de `tabindex` no necesitan ser consecutivos, pero el foco seguirá el orden establecido según estos números. Todos los controles de formulario pueden usar esta propiedad para personalizar el orden de navegación.

42- Formulario-Inhabilitat controles

Para inhabilitar controles en un formulario, se utiliza la propiedad `disabled`, que se puede aplicar a varios elementos como `button`, `input`, `optgroup`, `option`, `select`, y `textarea`.

Por ejemplo, deshabilitar un control de tipo `radio`:

```
<input type="radio" name="seccion" value="1" disabled>Niños
```

Esta propiedad se utiliza comúnmente en combinación con JavaScript, permitiendo habilitar o deshabilitar controles dinámicamente, según las interacciones del usuario con la página. Por ejemplo, podríamos mostrar tres secciones en un formulario, pero hacer que una de ellas no esté disponible para selección.

43- Formulario-text/password y maxlength

Los controles de tipo `"text"` y `"password"` tienen una propiedad llamada `"maxlength"` que limita la cantidad de caracteres que el usuario puede ingresar. Se asigna un valor entero que define el número máximo de caracteres permitidos. Es importante no confundir `"maxlength"` con `"size"`, ya que esta última solo controla la cantidad de caracteres visibles antes de que sea necesario hacer scroll.

44-Formulario-text/password/textarea y readonly

La propiedad "readonly" se puede aplicar a los controles de tipo "text", "password" y "textarea". Cuando se usa, el control se vuelve de solo lectura, lo que significa que el usuario no puede modificar su contenido. Esta propiedad es útil cuando un script cambia el estado del control en respuesta a ciertos eventos. Al contrario de "disabled", el control con "readonly" aún puede recibir foco, pero no se puede editar, mientras que con "disabled", el control no recibe foco y generalmente aparece deshabilitado visualmente.

45-Formulario-Envios de datos mediante mail

Para enviar los datos de un formulario por correo electrónico usando solo HTML, se utiliza la propiedad action con el valor mailto: seguido de la dirección de correo, como en:

```
<form action="mailto:pizzasya@html5.com" method="post"
enctype="text/plain">
```

Usaremos enctype="text/plain" para enviar los datos como texto plano, pero no se pueden adjuntar archivos.

46- Formulario-label

El elemento <label> se utiliza para asociar un texto con un control de formulario, lo que mejora la accesibilidad, especialmente para personas con discapacidad visual que usan programas de lectura de pantalla. En lugar de escribir el texto y el control por separado, se puede usar así:

```
<label for="nombre">Ingrese su nombre:</label>
<input type="text" name="nombre" size="20" id="nombre">
```

El atributo for del <label> hace referencia al id del control, lo que permite una mejor asociación entre ambos, facilitando la interacción y accesibilidad.

47- Frames

Antes se usaban para mostrar varios archivos HTML en una misma ventana del navegador, permitiendo la interacción entre ellos. Sin embargo, su uso dificulta la accesibilidad y la impresión de contenido, por lo que solo deberían utilizarse en casos muy específicos o cuando se mantengan sitios antiguos que los utilicen.

48- Frames-Actualización de un frame a partir del enlace de otro frame

Se solían utilizar para permitir la actualización de contenido en un frame a partir de un enlace en otro, y se podían hacer interactuar unos con otros. Tener en cuenta lo que hemos dicho antes sobre de que son antiguos, etc.

49-Frames-Asignación de medidas en píxeles

Es común asignar medidas a los frames, ya sea en porcentajes o en valores absolutos.

Cuando se usan porcentajes, como en `<frameset cols="20%,80%">`, los frames se redimensionan proporcionalmente al cambiar el tamaño de la ventana del navegador. Sin embargo, si se desea que un frame tenga un tamaño fijo, como el de un menú lateral que no debe cambiar, se puede utilizar un valor absoluto, por ejemplo: `<frameset cols="200px,*">`, para garantizar que ese frame se mantenga con un tamaño fijo mientras el otro se ajusta al resto del espacio.

50-Frames-Propiedades del elemento frame

es importante entender algunas propiedades adicionales del elemento `<frame>`:

- **noresize:** Al incluir esta propiedad, se evita que el frame sea redimensionado por el usuario con el mouse. Esto es útil, por ejemplo, en menús laterales.
- **frameborder:** Esta propiedad controla la visibilidad del borde del frame. Con un valor de 1, se muestra el borde; con 0, se oculta.

- **scrolling**: Define el comportamiento de las barras de desplazamiento dentro del frame. Los valores posibles son:
 - "auto": Muestra la barra de desplazamiento solo si el contenido no cabe en el frame.
 - "yes": La barra de desplazamiento siempre está visible.
 - "no": Nunca muestra la barra de desplazamiento.

51-Frames- anidamiento de frameset

Se puede anidar un <frameset> dentro de otro para crear una estructura de múltiples marcos. Esto se hace al incluir un frameset en el lugar de un frame, lo que permite dividir el área de la página en más de una sección.

Por ejemplo, si queremos crear una página con dos columnas, donde la segunda columna se divide en dos filas, podemos usar el siguiente enfoque:

```
<frameset cols="50%,50%">
  <frame src="columna1.html">
  <frameset rows="50%,50%">
    <frame src="fila1.html">
    <frame src="fila2.html">
  </frameset>
</frameset>
```

52-iframes

El elemento <iframe> en HTML permite incrustar un documento dentro de otro, funcionando de manera similar a una imagen, pero con contenido dinámico. Algunas de las propiedades útiles de un iframe son:

- **src**: Especifica el archivo que se mostrará dentro del iframe.
- **width**: Define el ancho del iframe en píxeles.
- **height**: Establece la altura del iframe en píxeles.
- **frameborder**: Controla si se muestra un borde alrededor del iframe. Puede ser 1 (borde visible) o 0 (sin borde).
- **scrolling**: Define el comportamiento de la barra de desplazamiento. Los valores pueden ser:

- "auto": El navegador decide si mostrarla.
- "yes": La barra de desplazamiento siempre es visible.
- "no": Nunca se muestra la barra de desplazamiento.
- **name**: Asigna un nombre al iframe, útil si se desea acceder o modificar su contenido desde otra página.

53-Declaración de DOCTYPE

La declaración DOCTYPE es esencial para definir el tipo de documento en una página HTML, lo que garantiza que se utilicen los estándares adecuados para la visualización en los navegadores. El W3C, que establece los estándares web, promueve el uso de directivas que aseguren la compatibilidad futura con nuevas versiones de navegadores.

La declaración DOCTYPE debe colocarse en la primera línea del archivo HTML, antes de la etiqueta <html>. En HTML5, la declaración DOCTYPE es simplificada y se utiliza la siguiente cadena:

```
<!DOCTYPE html>
```

Aunque lo ideal es usar HTML estricto para garantizar la correcta estructura y validación de las páginas, a veces se sigue utilizando HTML transicional, especialmente cuando se tiene un gran volumen de páginas antiguas que no son completamente compatibles con el HTML estricto. Esto permite una transición gradual hacia estándares más modernos

54-Declaración de DOCTYPE. HTML Transitional

La declaración DOCTYPE HTML Transitional se utiliza principalmente para el mantenimiento de sitios web antiguos. En este tipo de documento, se permite el uso de elementos HTML obsoletos relacionados con la presentación, como los de fuentes, alineación y colores, que en versiones más modernas de HTML están desaconsejados.

El HTML Transitional ofrece mayor flexibilidad en cuanto a la ubicación y el anidamiento de los elementos, lo que facilita la compatibilidad con páginas más antiguas. Sin embargo, se recomienda evitar su uso en nuevos proyectos, ya que existen estándares más modernos que favorecen una

mejor estructura y accesibilidad.

55-Declaración de DOCTYPE. HTML Estricto

La declaración DOCTYPE HTML Estricto excluye los elementos y atributos de presentación obsoletos, como aquellos relacionados con el diseño visual (fuentes, colores, alineación), que el W3C busca eliminar en favor del uso de hojas de estilo CSS. Se recomienda utilizar el DTD Estricto siempre que sea posible, ya que asegura una mayor coherencia y accesibilidad en la web.

Sin embargo, si es necesario soportar elementos de presentación antiguos, se puede recurrir al DTD Transicional. Para las páginas desarrolladas, puede ser necesario realizar ajustes para que cumplan con las validaciones del HTML estricto.

56-Declaración de DOCTYPE. Para Frames

Cuando se emplean frames en HTML, es necesario usar una declaración DOCTYPE específica. Para una página con frames, el DOCTYPE debe ser:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

En este caso, se utiliza la palabra "Frameset" en lugar de "Transitional" o dejarla vacía como en el HTML estricto.

57- Validación de la página a través de un enlace

Para validar una página HTML a través de un enlace, la W3C ofrece un ícono que puedes incluir en tu sitio. Este ícono es un enlace al validador de HTML que verifica si tu página cumple con los estándares. Solo necesitas agregar el código en un párrafo:

Ej:

```
<p>
```

```
<a href="http://validator.w3.org/check?uri=referer"></a>
```

```
</p>
```

58- Conceptos básicos de hojas de estilo

Las hojas de estilo en cascada (CSS) separan el contenido de la presentación en HTML. Mientras que HTML se encarga de mostrar el contenido, CSS define cómo se verá ese contenido. En las primeras versiones de HTML, se usaban elementos para mejorar la presentación, pero con la introducción de CSS, la presentación quedó completamente separada del contenido.

En CSS, cada elemento HTML tiene una propiedad llamada "style" que se utiliza para definir su apariencia. Las propiedades más comunes incluyen color (para el color del texto) y background-color (para el color de fondo). Estas propiedades se definen dentro de la etiqueta style, separadas por punto y coma.

Los colores en CSS se definen en formato hexadecimal (por ejemplo, #0000ff para azul). Es importante recordar que CSS es sensible a mayúsculas y minúsculas, por lo que color no es lo mismo que COLOR.

En resumen, CSS permite personalizar la apariencia de los elementos HTML, y es fundamental para mejorar el diseño y la accesibilidad de las páginas web.

59-Propiedades relacionadas al texto en CSS

En CSS, hay varias propiedades que se pueden aplicar al texto para modificar su apariencia y alineación:

- **text-align**: Alinea el texto a la izquierda, derecha, centro o justificado.
- **text-decoration**: Aplica efectos como subrayado, tachado, línea superior o ninguno.
- **letter-spacing**: Define el espacio entre caracteres.
- **word-spacing**: Ajusta el espacio entre palabras.
- **text-indent**: Indenta la primera línea de un párrafo, pudiendo usar valores negativos.
- **text-transform**: Controla la capitalización del texto (mayúsculas, minúsculas, o iniciales en mayúscula).

Estas propiedades permiten un control detallado sobre la apariencia del texto, mejorando la presentación en la página web.

60-Elementos HTML <div> y

Los elementos HTML <div> y se utilizan para agrupar otros elementos y aplicarles reglas de estilo.

- **<div>** es un contenedor de bloque, lo que significa que genera saltos de línea antes y después de su contenido, similar a las etiquetas <h1>, <p>, etc.
- **** es un contenedor en línea, por lo que no produce saltos de línea y se utiliza para aplicar estilo a pequeñas partes de texto o elementos dentro de una línea.

Ambos son más útiles cuando se combinan con CSS para dar estilo a los elementos agrupados.

61- Elemento HTML <pre>

El elemento HTML <pre> muestra el contenido tal como ha sido escrito, respetando los espacios en blanco y los saltos de línea. Por lo general, el texto dentro de esta etiqueta se muestra con una fuente monoespaciada, donde cada letra ocupa el mismo espacio en píxeles.

62-Propiedades relacionadas a las fuentes en CSS

Las propiedades CSS más importantes relacionadas con las fuentes son:

- **font-family:** Define la fuente a aplicar, como Arial, Courier New, Times New Roman, entre otras.
- **font-size:** Establece el tamaño de la fuente, pudiendo ser en píxeles.
- **font-style:** Permite definir el estilo de la fuente, con valores como normal, italic u oblique.
- **font-weight:** Determina el grosor de la fuente, usando valores como normal, bold, bolder o lighter.

63-Elementos y propiedades de HTML que no se deben usar

Existen elementos y propiedades de HTML que ya no deben utilizarse, ya que sus funciones se resuelven mejor con CSS. Algunos son:

Elementos:

- <u>, <s>, <strike>, <center>, <menu>, <dir>, <blink>, <marquee>, , , <i>, <big>, <small>, <hr>, <basefont>, <isindex>.

Propiedades:

- text, bgcolor, background, link, alink, vlink, align.

64- Subir las páginas HTML a un servidor de internet

Para subir una página HTML a un servidor de Internet, primero debemos gestionar un dominio y un servicio de hosting. Si es solo para aprender, podemos usar un hosting gratuito, como 000webhost, que nos permitirá gestionar un subdominio y subir nuestras páginas mediante FTP.

Los pasos son:

1. **Registrar una cuenta en un hosting gratuito**, por ejemplo, 000webhost, y crear un subdominio.
2. **Obtener los datos FTP** de conexión desde el panel de control del hosting.
3. **Instalar un programa FTP** como FileZilla y configurarlo con los datos obtenidos.
4. **Conectar** FileZilla al servidor utilizando los datos FTP proporcionados por la empresa de hosting.
5. **Subir los archivos** desde tu computadora a la carpeta public_html en el servidor.
6. **Subir una página index.html** para que se cargue de forma predeterminada al acceder al sitio.

Así, cualquiera con acceso a Internet podrá visualizar estas páginas HTML.