meta:>

<meta name=viewpport-> tamaño de la ventana  
5. Etiquetas de secciones-> xq etiquetas html 5 pa diferenciarlos semanticmente

adrees -> direcion pag web o coreo se pone abajo

profecto nav aside,articule,meta poryecti qno coriige

html 5 evaluar con sas una parte ccss5 y html y sas  
etiqueta time web confierto ej > si no sabes qn hace concierto buscas x fecha x eso hay q decir q fecha  
ej  
La etiqueta <time> y <data>

Las fechas y horarios son partes fundamentales de los contenidos de las páginas

web. Los motores de búsqueda son capaces de filtrar los resultados basándose en el

tiempo y, en algunos casos, un resultado de búsqueda específico puede recibir más o

menos peso en función de cuándo fue publicado por primera vez. El elemento <time>

representa un período específico en el tiempo. Puede incluir el atributo datetime para

convertir las fechas en un formato interno legible por un ordenador.

La etiqueta <time> en HTML5 se utiliza para **especificar una fecha y/o hora** de una manera semántica, es decir, que el navegador y los motores de búsqueda puedan entender y procesar esa información de forma más eficiente.

<p>El concierto fue el <time datetime="2001-05-15T19:00">15 de Mayo</time>.</p>  
  
ej .-><https://www.w3schools.com/tags/att_time_datetime.asp>

su tenog una fase q la quiero corarw->   
La etiqueta

, y El elemento

produce un salto del línea en el texto (retorno de carro) Es útil para escribir un poema o dirección donde la división de las líneas es significante.

IES San Vicente  
 Calle Lillo Juan  
 San Vicente del Raspeig

La etiqueta funciona de forma similar a la anterior etiqueta, pero a diferencia de esta, se utiliza para evitar que se desborde el contenido, cortando la palabra en el caso de poder hacerlo, de lo contrario permanecerá junta. De tal forma que si el contenedor que tiene la el siguiente ejemplo es muy pequeño la palabra supernova sea cortada.

supernova

Por otro lado, la etiqueta representa un cambio de tema entre párrafos. En versiones previas de HTML se representaba una línea horizontal, todavía puede representarse como una línea en los navegadores visuales, pero ahora es definida en términos semánticos y no tanto en términos representativos, por tanto para representar una línea en horizontal deberíamos utilizar CSS.

Este es un párrafo únicamente visual para ver el uso de la etiqueta hr

Este es el segundo párrafo, separado del primero por la etiqueta hr, que puede representarse mediante una línea Rosa Medina Diseño de Interfaces Web Página 21/49 Tema 1: HTML y CSS (1ª Parte) horizontal.

Diferencia entre las etiquetas <br> y <wbr> en HTML

Ambas etiquetas se utilizan para controlar el salto de línea en el texto HTML, pero tienen funciones y usos muy diferentes:

<br> (Salto de línea forzado)

Función: Inserta un salto de línea de forma obligatoria en el punto donde se coloca.

Uso: Se utiliza cuando se necesita un salto de línea en un lugar específico, sin importar el tamaño de la pantalla o el contenedor. Por ejemplo:

En direcciones postales para separar la calle del número.

En poemas o letras de canciones para respetar la métrica.

En listas sin viñetas para separar los elementos.

Ejemplo:

HTML

<p>Esta es la primera línea.<br>Esta es la segunda línea.</p>

Usa el código con precaución.

<wbr> (Oportunidad de salto de línea)

Función: Indica al navegador que en ese punto es posible realizar un salto de línea si es necesario para evitar que una palabra se quede cortada al final de una línea.

Uso: Se utiliza para mejorar la apariencia del texto en pantallas pequeñas o cuando se tienen palabras muy largas que podrían romper el diseño de la página. Por ejemplo:

En palabras compuestas muy largas.

En URLs o direcciones de correo electrónico.

Ejemplo:

HTML

<p>Esta<wbr>palabra</wbr>esmuy<wbr>larga.</p>

Usa el código con precaución.

Resumen de las diferencias:

Característica <br> <wbr>

Función Salto de línea obligatorio Oportunidad de salto de línea

Uso Cuando se necesita un salto de línea en un lugar específico Para evitar que las palabras se corten

Comportamiento Siempre inserta un salto de línea Inserta un salto de línea solo si es necesario

Exportar a Hojas de cálculo

Cuándo usar cada una:

Utiliza <br> cuando necesites un control preciso sobre dónde se producen los saltos de línea.

Utiliza <wbr> cuando quieras mejorar la apariencia del texto y evitar que las palabras se corten de forma antiestética.

En resumen:

<br> es como poner un punto y seguido a la fuerza.

<wbr> es como decirle al navegador: "Si la palabra no cabe aquí, puedes cortarla".

Ejemplo práctico:

Imagina una dirección de correo electrónico muy larga: micorreoelectronico@dominioextremadamentelargo.com. Si no utilizamos <wbr>, es posible que esta dirección se corte de forma poco estética en la pantalla. Al agregar <wbr> después de cada palabra, permitimos que el navegador divida la dirección en varias líneas si es necesario.

HTML

<p>Mi correo electrónico es: <wbr>micorreoelectronico@<wbr>dominioextremadamentelargo.com</p>

Usa el código con precaución.

Conclusión:

Tanto <br> como <wbr> son herramientas útiles para controlar la apariencia del texto en una página web. La elección de una u otra dependerá del efecto que se quiera conseguir y de las características del contenido.

kdb .>temas y   
a etiqueta <kbd> se utiliza precisamente para representar la entrada de datos por parte del usuario. El nombre de la etiqueta proviene de keyboard, ya que tradicionalmente siempre se ha utilizado el teclado para proporcionar información por parte del usuario.

El ejemplo más común y fácil de entender es la entrada de datos mediante combinaciones de teclado para realizar una acción determinada.

Veamos un ejemplo práctico de la etiqueta <kbd>. Necesitamos indicar al lector que al pulsar una cierta combinación de teclas, podrá abrir una nueva pestaña en el navegador.  
<p>

Para abrir una nueva pestaña en el navegador,

pulsa la combinación de teclas <kbd>CTRL</kbd>+<kbd>T</kbd>.

</p>

La etiqueta <samp> es la opuesta a la etiqueta <kbd>. Si la última representa la entrada de datos por parte del usuario a la máquina, la etiqueta <samp> nos permite representar la salida de la información que proviene de una máquina, hacia el usuario.

Mediante esta etiqueta podemos mostrar la salida de un programa o sistema a un usuario:

<p>

<strong>Internet Explorer</strong> mostró un mensaje: <samp>Error: Todo ha ido bien</samp>.

</p>

De esta forma, podemos indicar tanto entrada como salida de datos desde una máquina.

./-> haxe referencia donde estas

. Ruta absoluta y ruta relativa Si estamos desarrollando nuestra aplicación en un directorio llamado lenguajes que tiene el siguiente contenido:

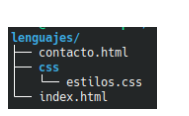
Dentro del atributo href debemos indicar la ruta a la web que queremos acceder al hacer clic. Esta ruta la podemos indicar de dos modos:

1. Como una ruta absoluta a la raíz de la aplicación. Como contacto.html está en la carpeta raíz de la aplicación lenguajes, el enlace quedaría así: Contacta con nosotros

<a href="/contacto.html" title="Página de contacto">Contacta con nosotros</a>

2. Como una ruta relativa desde la carpeta en donde nos situamos. Por ejemplo si nos encontramos en la misma carpeta el enlace sería:

<a href="contacto.html" title="Página de contacto">Contacta con nosotros</a>

a) En el caso de estar dentro de una subcarpeta y debemos subir de nivel (acceder a la carpeta padre) debemos utilizar .. pudiendo enlazarlos para subir de nivel en varias carpetas, por ejemplo ../../../ para salir de tres carpetas. Contacta con nosotros  
<a href="../contacto.html" title="Página de contacto">Contacta con nosotros</a>  
  
8.1. Imágenes <img>

Para insertar imágenes en nuestra web utilizamos la etiqueta <img> con su atributo src

para indicar en dónde se encuentra la imagen. La etiqueta <img> es un elemento vacío, por

lo que no es necesario que introduzcamos la etiqueta de cierre. Podemos utilizar 3

formatos de imágenes para nuestra página: JPEG, GIF y PNG.

• Con JPEG conseguimos el menor tamaño, aunque no la mejor calidad. Está

recomendado para imágenes medianas o grandes con muchos colores y variedad.

No soporta transparencias.

• El formato GIF soporta transparencias, pero está muy limitado en colores (256).

También nos permite animaciones sencillas.

• El formato PNG es el que mejor calidad nos da, además de soportar

transparencias. Todo esto a un coste un poco mayor en cuanto tamaño.

• Formato SVG es el formato vectorial, este formato es el ideal para imágenes

escalables

Se debe poner el atributo alt a todas las imágenes, establece un texto alternativo que

describa la imagen a mostrar.

<img src="../imgs/fondo.png" alt="fondo de la web">

Rosa Medina Diseño de Interfaces Web Página 23/49

Tema 1: HTML y CSS (1ª Parte)

Además de estos dos atributos, la etiqueta <img> admite los siguientes atributos:

• width: ancho de la imagen (en píxeles o porcentaje). Si no indicamos altura,

mantiene proporciones de la imagen. Normalmente se establece por CSS

• height: alto de la imagen.

calidad transparencias

|  | escalable | calidad | tamaño | transparencia | Vectorial | Mucho color |
| --- | --- | --- | --- | --- | --- | --- |
| jpg |  | no mejor | menor | no soporta |  | recom img medianas o grandes muchos colorea |
| png |  | Mayor |  | soporta |  |  |
| Gif |  |  |  | soopotra |  |  |
| SVG | SVG |  |  |  | formato vectorial |  |

Con JPEG conseguimos el menor tamaño, aunque no la mejor calidad. Está recomendado para imágenes medianas o grandes con muchos colores y variedad. No soporta transparencias. • El formato GIF soporta transparencias, pero está muy limitado en colores (256). También nos permite animaciones sencillas. • El formato PNG es el que mejor calidad nos da, además de soportar transparencias. Todo esto a un coste un poco mayor en cuanto tamaño. • Formato SVG es el formato vectorial, este formato es el ideal para imágenes escalables  
  
  
Aquí tienes una tabla que resume las características de los formatos de imagen **JPG**, **PNG**, **GIF**, y **SVG** en relación a los criterios que mencionaste:

| **Caracterítica** | **JPG** | **PNG** | **GIF** | **SVG** |
| --- | --- | --- | --- | --- |
| **Escalable** | No escalable (raster) | No escalable (raster) | No escalable (raster) | Escalable (vectorial) |
| **Calidad** | Alta, pero puede perder calidad al comprimir | Alta, sin pérdida de calidad | Calidad limitada (256 colores) | Alta (sin pérdida de calidad) |
| **Tamaño** | Tamaño de archivo pequeño a moderado, depende de la compresión | Tamaño de archivo más grande que JPG | Tamaño pequeño a moderado | Tamaño de archivo generalmente pequeño |
| **Transparencia** | No soporta transparencia | Soporta transparencia | Soporta transparencia limitada | Soporta transparencia completa |
| **Vectorial** | No (raster) | No (raster) | No (raster) | Sí (vectorial) |
| **Mucho color** | Soporta millones de colores | Soporta millones de colores | Soporta hasta 256 colores | Soporta millones de colores |

### **Notas adicionales:**

* **JPG**: Ideal para fotografías y imágenes complejas donde se requiere un buen equilibrio entre calidad y tamaño de archivo.
* **PNG**: Recomendado para gráficos, logotipos y cualquier imagen que requiera fondo transparente o alta calidad.
* **GIF**: Usado principalmente para animaciones cortas y gráficos simples con una paleta de colores limitada.
* **SVG**: Perfecto para gráficos, logotipos y cualquier diseño que necesite escalar sin perder calidad, como iconos y gráficos vectoriales.

Si necesitas más información sobre algún formato específico o un tema relacionado, ¡háznoslo saber!

sprite se meten todos olo q tienes

| **Término/Formato** | **Descripción** | **Uso en Diseño** |
| --- | --- | --- |
| **Escalable** | Capacidad de una imagen para aumentar o disminuir su tamaño sin perder calidad. | Esencial para que las imágenes se vean bien en diferentes dispositivos y resoluciones. |
| **Calidad** | Claridad y detalle de una imagen; puede ser afectada por el formato y la compresión. | Mantener alta calidad es crucial para que los elementos visuales sean atractivos y profesionales. |
| **Tamaño** | Dimensiones de una imagen (ancho x alto) y tamaño del archivo (peso). | Crucial para el rendimiento del sitio web; imágenes grandes pueden ralentizar la carga de la página. |
| **Transparencia** | Capacidad de una imagen de mostrar áreas transparentes, permitiendo que el fondo se vea a través. | Útil para crear superposiciones, efectos visuales y para integrar mejor las imágenes en el diseño. |

### 

### 

### **Formatos de Imagen**

| **Formato** | **Características** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| **JPG** | Formato con compresión con pérdida, ideal para fotografías. | Tamaño de archivo pequeño; buena calidad para fotos. | Pérdida de calidad al guardar varias veces; no admite transparencia. |
| **PNG** | Formato sin pérdida que admite transparencias. Ideal para gráficos y logotipos. | Alta calidad y soporte de transparencia. | Tamaño de archivo más grande que JPG. |
| **GIF** | Formato limitado a 256 colores; soporta animaciones. | Soporte para animaciones y transparencias. | Limitación de colores; calidad baja para fotos. |
| **SVG** | Formato vectorial, escalable sin pérdida de calidad. Ideal para gráficos y logotipos. | Escalable sin perder calidad; archivos pequeños para gráficos simples. | Puede ser complicado para gráficos muy complejos. |

Esta tabla resume de manera clara los conceptos y características que mencionaste, facilitando su comprensión y comparación. Si necesitas más información sobre algún término o formato específico, ¡no dudes en preguntar!

canal: 64 b-> bits

640 b / en 2 seg \_> 320bs

quiere decir esto:

640b / 64 b = 10 veces-> el periodo 10 veces en 2 seg

10/2 = 5 veces !! \_> hace 5 envios q es lel periodo

frecuenci = 1/T = (1/5 )/seg = (0.2 seg ) /seg

en 1 segundo haces 0,2 -< haces un envío –caca

1 envio en 0,2 segundos->en 5 envios haces 1 seg y en 2seg 10 envio

la frecuencia es lo opuesto del periodo  
  
explicacipn  
**Contexto de la Explicación**

Parece que estamos hablando sobre la transmisión de datos en un canal de comunicación donde se están enviando bits (unidades de información).

### **Desglose de Cálculos y Conceptos**

1. **Canal: 64 b → bits**
   * Esto significa que el canal tiene un ancho de banda de 64 bits. Este es el máximo de datos que puede enviar el canal en un momento dado.
2. **640 b / en 2 seg → 320 b/s**
   * Esto indica que en un período de 2 segundos, el canal transmite un total de 640 bits.
   * Por lo tanto, la tasa de transmisión es: 640 bits2 segundos=320 bits por segundo (b/s)\frac{640 \text{ bits}}{2 \text{ segundos}} = 320 \text{ bits por segundo (b/s)}2 segundos640 bits​=320 bits por segundo (b/s)
3. **640 b / 64 b = 10 veces → el periodo 10 veces en 2 seg**
   * Aquí se está calculando cuántas veces puede enviarse un bloque de 64 bits dentro de los 640 bits.
   * Si divides 640 bits entre 64 bits, obtienes: 640 bits64 bits=10\frac{640 \text{ bits}}{64 \text{ bits}} = 1064 bits640 bits​=10
   * Esto significa que en 2 segundos se pueden enviar 10 bloques de 64 bits.
4. **10/2 = 5 veces → hace 5 envíos que es el periodo**
   * Si tienes 10 envíos en 2 segundos, el número de envíos por segundo (frecuencia) es: 10 envıˊos2 segundos=5 envıˊos por segundo\frac{10 \text{ envíos}}{2 \text{ segundos}} = 5 \text{ envíos por segundo}2 segundos10 envıˊos​=5 envıˊos por segundo
   * Así que, se envían 5 bloques de 64 bits por segundo.
5. **Frecuencia = 1/T = (1/5)/seg = (0.2 seg)/seg**
   * La frecuencia fff se calcula como el recíproco del período TTT (tiempo por ciclo de envío): f=1Tf = \frac{1}{T}f=T1​
   * Aquí, si TTT es el tiempo entre cada envío, y ya hemos calculado que hay 5 envíos por segundo, entonces: T=15 segundos=0.2 segundosT = \frac{1}{5} \text{ segundos} = 0.2 \text{ segundos}T=51​ segundos=0.2 segundos
   * Esto significa que el tiempo entre cada envío es de 0.2 segundos.
6. **1 envío en 0.2 segundos → en 5 envíos haces 1 seg y en 2 seg 10 envíos**
   * Esto se refiere a que si cada envío toma 0.2 segundos, en 1 segundo podrás realizar 5 envíos (ya que 1 seg0.2 seg/envıˊo=5 envıˊos\frac{1 \text{ seg}}{0.2 \text{ seg/envío}} = 5 \text{ envíos}0.2 seg/envıˊo1 seg​=5 envıˊos).
   * De manera similar, en 2 segundos, podrás hacer: 2 seg×5 envıˊos/seg=10 envıˊos2 \text{ seg} \times 5 \text{ envíos/seg} = 10 \text{ envíos}2 seg×5 envıˊos/seg=10 envıˊos

### **Resumen**

* **Tasa de Transmisión**: El canal tiene una capacidad de 64 bits y puede enviar 640 bits en 2 segundos, lo que da como resultado 320 bits por segundo.
* **Envíos por Segundo**: Puedes enviar 10 bloques de 64 bits en 2 segundos, que equivale a 5 bloques por segundo.
* **Tiempo por Envío**: Cada envío toma 0.2 segundos, lo que significa que en 1 segundo se pueden hacer 5 envíos y en 2 segundos, 10 envíos.

Este análisis te permite entender cómo funciona la transmisión de datos en un canal y cómo calcular la tasa de envío y el tiempo por envío. Si tienes más preguntas sobre estos conceptos o necesitas más aclaraciones, ¡no dudes en preguntar!

8.1 usarlños  
pictue

8.1.4.Nuevas etiquetas de imágenes

HTML5.1 incorpora nuevas etiquetas de imágenes además de la etiqueta que había

antes (<img>):

Etiqueta Atributos Descripción

<picture> Agrupa una serie de imágenes. Es una etiqueta

contenedora

<source> srcset, sizes, media, type Muestra la imagen que cumpla una serie de

criterios opcionales

Como podemos ver, la etiqueta <source> tiene una serie de atributos disponibles para

utilizar:

• srcset: varas imágenes separadas por comas. (atributo obligatorio)

• sizes: Tamaño con el que queremos que se visualice la imagen

• media: Condición (media queries, CSS) a cumplir para que la imagen se muestre.

• type: Tipo de formato de imagen (Atributo opcional)

Una de las ventajas que tiene estas etiquetas es que podemos utilizar diferentes

formatos de la imagen dependiendo si el navegador soporta ese formato o no.

<picture>

<source srcset="imagen.webp"><!-- Formato WebP -->

<source srcset="imagen.jxr"><!-- Formato JPEG XR -->

<img src="imagen.jpg" alt="Descripción de la imagen">

</picture>

En el ejemplo anterior, lo que estamos diciendo es que utilice primero la imagen en

formato WebP, si no soporta este formato pasará a mostrar el formato JPEG XR, pero si

tampoco lo soporta mostrará entonces la imagen JPEG que es soportada por todos los

navegadores a excepción de aquellos que no muestran imágenes como Lynx que

mostrará el texto alternativo (alt).

Utilizando el atributo media, podemos crear imágenes responsive que cambien

dependiendo del min-width/max-width de la pantalla.

<picture>

<source media="(min-width: 600px)" srcset="imagen-xl.png">

<source media="(min-width: 300px) and (max-width: 600px)" srcset="imagen-large.png">

<source media="(max-width: 50px)" srcset="imagen-small.png">

<img src="imagen-medium.png" alt="HTML5 logo">

</picture>

del ancho en concreto) se muestre una imagen u otra:

• Dispositivos con una resolución de pantalla mayor a 600px: imagen-xl.png

• Dispositivos con una resolución entre 300 y 600px: imagen-large.png

• Dispositivos con una resolución menor a 50px: imagen-small.png

• Dispositivos que no cumple las condiciones anteriores o no soporta HTML5.1:

imagen-medium.png

Por último, esta etiqueta nos permite indicar diferentes imágenes dependiendo la

densidad de la pantalla (alto). Para ello, tenemos que usar un descriptor tras el nombre de

la imagen con el atributo srcset(el valor por defecto es 1x)

<picture>

<source media="(min-width: 600px)" srcset="imagen-xl.png, imagen-xl-hd.png 2x, imagen-xl-fhd.png 3x" />

<source media="(min-width: 300px) and (max-width: 600px)" srcset="imagen-large.png, imagen-large-hd.png 2x,

imagen-large-fhd.png 3x" />

<source media="(max-width: 50px)" srcset="imagen-small.png, imagen-small-hd.png 2x, imagen-small-fhd.png 3x" />

<img srcset="imagen-medium.png, imagen-medium-hd.png 2x, imagen-medium-fhd.png 3x" alt="HTML5 logo" />

</picture>

cuanto mas spequelo mas denso mas grande menos denso  
El elemento <picture> es una forma de manejar imágenes responsivas en HTML5. Permite especificar diferentes fuentes de imágenes dependiendo de las características de la pantalla (como el tamaño de la ventana del navegador).

Componentes del Código

<source>:

Se utiliza dentro del elemento <picture> para definir diferentes fuentes de imagen según las condiciones de medios especificadas.

Cada <source> tiene un atributo media, que establece la condición que debe cumplirse para usar esa imagen. Por ejemplo, (min-width: 600px) significa que la imagen se utilizará si el ancho de la ventana es de al menos 600 píxeles.

srcset:

El atributo srcset se utiliza para proporcionar múltiples versiones de una imagen. Permite al navegador seleccionar la imagen más adecuada en función de la resolución de la pantalla y las condiciones del medio.

Ejemplo: srcset="imagen-xl.png, imagen-xl-hd.png 2x, imagen-xl-fhd.png 3x"

imagen-xl.png: Esta es la imagen que se usará por defecto si la condición del media se cumple.

imagen-xl-hd.png 2x: Indica que esta versión de la imagen es de mayor resolución, adecuada para pantallas con una densidad de píxeles de 2x (como la mayoría de los dispositivos de pantalla retina).

imagen-xl-fhd.png 3x: Similarmente, esta imagen está diseñada para pantallas con una densidad de píxeles de 3x, ofreciendo aún más detalle.

1x, 2x, 3x:

Estos números indican la densidad de píxeles de la pantalla en la que la imagen debe ser utilizada:

1x: Es la resolución estándar. Se refiere a pantallas con una densidad de píxeles normal.

2x: Se refiere a pantallas de alta resolución (como las pantallas Retina de Apple). Estas pantallas tienen el doble de píxeles que las pantallas estándar en el mismo espacio físico.

3x: Indica pantallas con una densidad de píxeles aún mayor, donde hay tres veces más píxeles que en una pantalla estándar.

¿Cómo funciona?

Cuando un navegador encuentra el elemento <picture>, evalúa las condiciones de cada <source> en orden. Cuando encuentra una condición que se cumple, selecciona la imagen adecuada del srcset correspondiente.

Si el navegador no encuentra ninguna condición que se cumpla, utilizará la imagen del <img> como última opción.

Por ejemplo:

Si el ancho de la ventana es de 800 píxeles (cumple con min-width: 600px), el navegador podría elegir imagen-xl-hd.png si está en un dispositivo de 2x de resolución, o imagen-xl-fhd.png en un dispositivo de 3x.

Ventajas de usar <picture> y srcset

Carga eficiente: Solo se cargan las imágenes necesarias para la pantalla actual, lo que ahorra ancho de banda y mejora la velocidad de carga.

Adaptabilidad: Permite que el diseño se vea bien en cualquier dispositivo, independientemente de su tamaño o resolución de pantalla.

Resumen

El uso de <picture> junto con srcset permite ofrecer diferentes imágenes según las características del dispositivo y la densidad de píxeles, mejorando la calidad visual y la eficiencia del uso de recursos en un sitio web. Si tienes más preguntas sobre este tema o cualquier otro, ¡no dudes en preguntar!

El código que has compartido utiliza la etiqueta <picture> de HTML5, que es una forma avanzada de manejar imágenes responsivas. Vamos a desglosar el código para entender cómo funciona y qué hace cada parte.

### **Estructura del Código**

<picture>

<source media="(min-width: 600px)" srcset="imagen-xl.png, imagen-xl-hd.png 2x, imagen-xl-fhd.png 3x" />

<source media="(min-width: 300px) and (max-width: 600px)" srcset="imagen-large.png, imagen-large-hd.png 2x, imagen-large-fhd.png 3x" />

<source media="(max-width: 50px)" srcset="imagen-small.png, imagen-small-hd.png 2x, imagen-small-fhd.png 3x" />

<img srcset="imagen-medium.png, imagen-medium-hd.png 2x, imagen-medium-fhd.png 3x" alt="HTML5 logo" />

</picture>

### **Explicación por Partes**

1. **Etiqueta <picture>**:
   * Esta etiqueta se utiliza para contener varias fuentes de imágenes. Es especialmente útil para hacer que las imágenes sean responsivas, es decir, que se ajusten a diferentes tamaños de pantalla.
2. **Etiquetas <source>**:
   * Dentro de <picture>, puedes tener múltiples etiquetas <source>. Cada etiqueta <source> especifica condiciones de medios (media queries) y un conjunto de imágenes a usar (srcset).
   * **Atributo media**: Define las condiciones bajo las cuales se debe usar la imagen especificada en el atributo srcset. Estas condiciones se basan en el tamaño de la pantalla del dispositivo.
3. **Atributo srcset**:
   * Este atributo especifica una lista de imágenes que se pueden usar dependiendo de la densidad de píxeles del dispositivo (1x, 2x, 3x, etc.).
   * La sintaxis es:
     + **imagen.png**: URL de la imagen.
     + **imagen-hd.png 2x**: URL de la imagen de alta definición que se utilizará si el dispositivo tiene una densidad de píxeles de 2x.
     + **imagen-fhd.png 3x**: URL de la imagen de calidad superior que se utilizará si el dispositivo tiene una densidad de píxeles de 3x.

Condiciones Específicas en Tu Código

Primera etiqueta <source>:

<source media="(min-width: 600px)" srcset="imagen-xl.png, imagen-xl-hd.png 2x, imagen-xl-fhd.png 3x" />

Condición: Si la pantalla tiene un ancho mínimo de 600 píxeles.

Imágenes Usadas: imagen-xl.png, imagen-xl-hd.png para pantallas de alta densidad (2x), y imagen-xl-fhd.png para pantallas de muy alta densidad (3x).

Segunda etiqueta <source>:

<source media="(min-width: 300px) and (max-width: 600px)" srcset="imagen-large.png, imagen-large-hd.png 2x, imagen-large-fhd.png 3x" />

Condición: Si la pantalla tiene un ancho entre 300 y 600 píxeles.

Imágenes Usadas: imagen-large.png, imagen-large-hd.png para pantallas de alta densidad, y imagen-large-fhd.png para pantallas de muy alta densidad.

Tercera etiqueta <source>:

<source media="(max-width: 50px)" srcset="imagen-small.png, imagen-small-hd.png 2x, imagen-small-fhd.png 3x" />

Condición: Si la pantalla tiene un ancho máximo de 50 píxeles.

Imágenes Usadas: imagen-small.png, imagen-small-hd.png para pantallas de alta densidad, y imagen-small-fhd.png para pantallas de muy alta densidad.

Etiqueta <img>:

<img srcset="imagen-medium.png, imagen-medium-hd.png 2x, imagen-medium-fhd.png 3x" alt="HTML5 logo" />

Si ninguna de las condiciones anteriores se cumple, se utilizará esta imagen como último recurso.

Imágenes Usadas: imagen-medium.png, imagen-medium-hd.png para pantallas de alta densidad, y imagen-medium-fhd.png para pantallas de muy alta densidad.

alt: Proporciona un texto alternativo para la imagen en caso de que no se cargue o para lectores de pantalla.

Resumen de Funcionalidad

Responsividad: El código permite que el navegador seleccione la imagen más apropiada para el tamaño de pantalla y la densidad de píxeles del dispositivo del usuario. Esto mejora la carga de la página al garantizar que solo se descargue la imagen necesaria.

Optimización: Al usar diferentes tamaños de imagen, se optimiza el rendimiento y la calidad visual en diferentes dispositivos, asegurando que la experiencia del usuario sea óptima sin importar cómo accedan a tu contenido.

Si tienes alguna otra pregunta o necesitas más aclaraciones, ¡estaré encantado de ayudarte!

<https://htmldesdecero.es/atributos/srcset/>

audio y video en los proyectos

pag web ejemplo musici al `rimcip  
pagina 26

8.4. Etiqueta <track>

Con HTML5 se añade el estándar webVTT, destinado a la creación de subtítulos

en archivos de vídeo o audio. Este formato se puede utilizar tanto dentro de la etiqueta

<video> como de <audio>. Los atributos que puede contener son: src, scrlang (codigo del

idioma), label (título que verá al pulsar el botón CC), default (booleano que marca como los

subtítulos principales por defecto).

<video width="640" height="480">

<source src="video.mp4" type="video/mp4">

<source src="video.webm" type="video/webm">

<source src="video.ogv" type="video/ogg">

<img src="imagen.png" alt="Video no soportado">

<track src="sub-es.vtt" srclang="es" label="Español" default>

<track src="sub-en.vtt" srclang="en" label="English" default>

<track src="sub-fr.vtt" srclang="fr" label="Français" default>

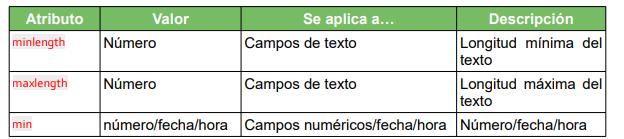
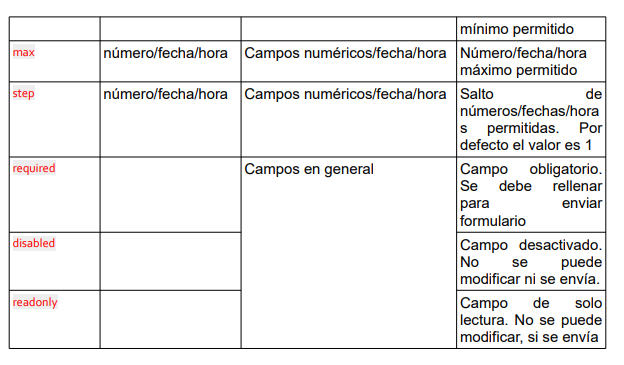
</video>

Como web de refuerzo tenéis <https://htmldesdecero.es/etiquetas/track/>

estandar es una norma

TABLAS NOOOO importante

11.4 el input tye number y range quiere  
Ç

12. Validación de formularios Cuando creamos un formulario debemos tener en cuenta que los usuarios se pueden equivocar a la hora de rellenar nos datos del mismo, y por tanto, debemos anticiparnos a estos e intentar que los datos lleguen correctamente al servidor sin ningún tipo de error. Para evitar que los datos lleguen con errores debe haber un proceso de validación, esta validación se debe realizar tanto en la parte front-end (cliente, JS) como en la parte back-end(servidor, PHP). Tradicionalmente la validación de un formulario siempre la ha hecho JavaScript, pero HTML5 introduce nuevos atributos que nos permiten validar los campos de un formulario. Estos atributos que nos permiten validar un formulario son:  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Además de estos atributos, tenemos una serie de patrones para validar los campos en HTML utilizando expresiones regulares para validar los datos. Una expresión regular no es más que una cadena que representa un posible patrón de coincidencias. Si queremos utilizar un patrón, deberemos utilizar el atributo pattern y como valor la expresión regular que deba cumplir  
