

Propiedades HTML y CSS

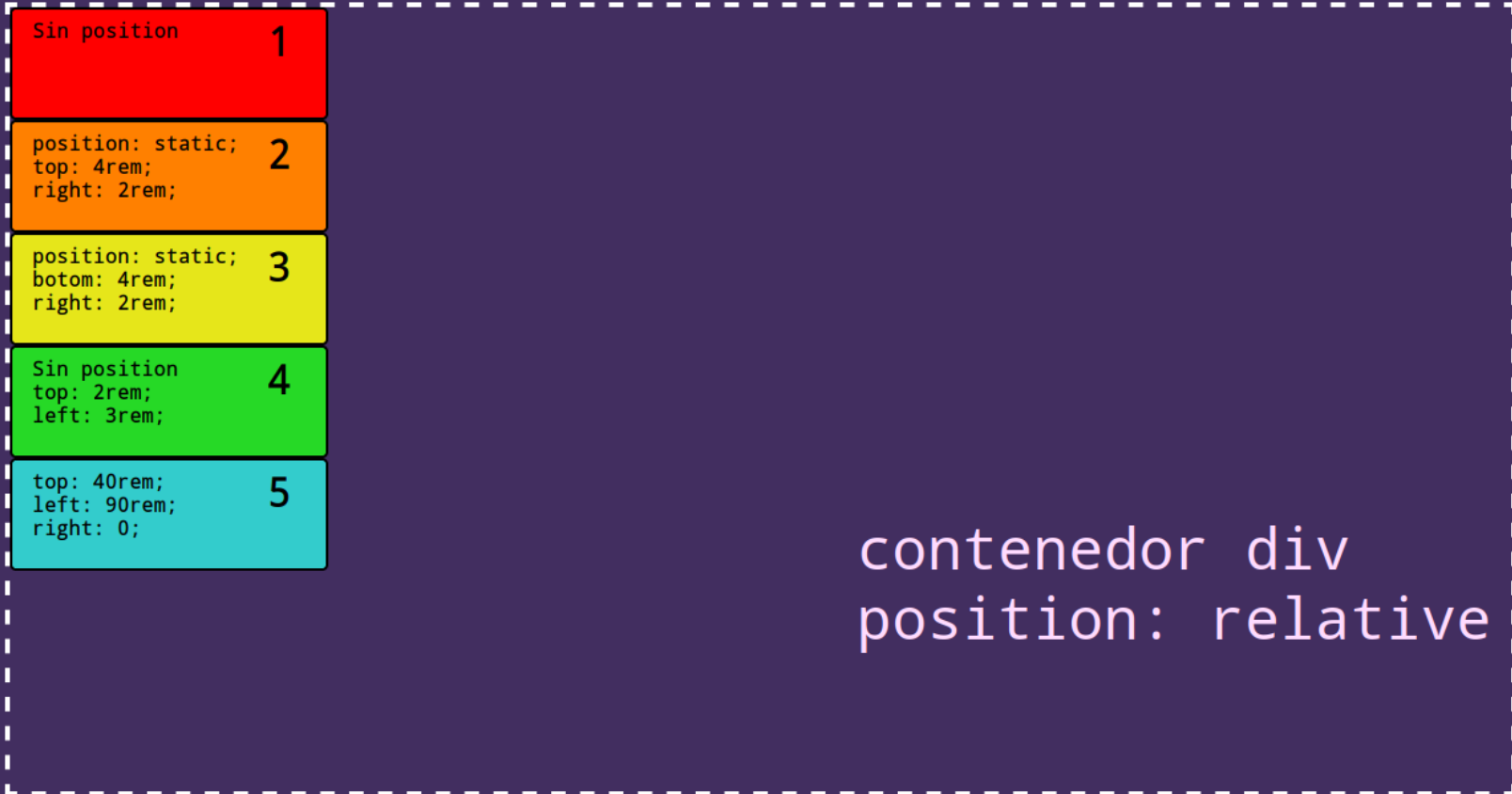
//Nahia //Allan //Moisés //Silvia

Position

position

- static** Valor por defecto
- relative** El elemento se coloca relativo al flujo normal
- absolute** El elemento se coloca respecto a su contenedor posicionado más cercano
- fixed** El elemento se coloca respecto a la ventana del navegador
- sticky** Es una combinación entre relative y fixed

position: static



position: relative

```
position: relative; 1  
top: 0rem;  
left: 0rem;
```

```
position: relative; 2  
bottom: 1rem;  
left: 2rem;
```

```
position: relative; 3  
left: 0rem;
```

```
position: relative; 4  
bottom: 2rem;  
left: 20rem;
```

```
position: static; 5  
top: 5rem;  
left: 8rem;
```

contenedor div
position: relative

position: absolute

```
position: relative; 5  
top: 0rem;  
left: 0rem;
```

```
position: absolute; 1  
top: 0rem;  
right: rem;
```

```
position: absolute; 3  
top: 50%;  
left: 50%;
```

contenedor div
position: relative

```
position: absolute; 4  
bottom: 0rem;  
left: 0rem;
```

```
position: absolute; 2  
bottom: 0rem;  
right: 0rem;
```

position: fixed

Position — Mozilla Firefox

Position

127.0.0.1:5500/position/fixed.html

```
position: absolute;
top: 0rem;
right: rem;
```

1

```
position: absolute;
top: 50%;
left: 50%;
```

3

position: fixed

Ejemplo:

```
position: absolute;
bottom: 0rem;
right: 0rem;
```

2

```
position: relative;
top: 0rem;
left: 0rem;
```

5

```
position: absolute;
bottom: 0rem;
left: 0rem;
```

4

Formularios

Formularios <form>

Los formularios en HTML son elementos clave para recoger datos del usuario y enviarlos al servidor para su procesamiento.

```
<form action="ejemplo.php" method="get">
  <p>Nombre: <input type="text" name="nombre" size="40"></p>
  <p>Año de nacimiento: <input type="number" name="nacido" min="1900"></p>
  <p>Sexo:
    <input type="radio" name="hm" value="h"> Hombre
    <input type="radio" name="hm" value="m"> Mujer
  </p>
  <p>
    <input type="submit" value="Enviar">
    <input type="reset" value="Borrar">
  </p>
</form>
```

Nombre:

Año de nacimiento:

Sexo: ☐ Hombre ☐ Mujer

Enviar

Borrar

Atributos más importantes

action: contiene el nombre del agente que procesará los datos remitidos al servidor (por ejemplo, un script de PHP)

method: define la manera de enviar los datos al servidor.

get: los valores enviados se añaden a la dirección indicada en el atributo action.

post: los valores se envían de forma separada.

“Si el atributo method no está establecido, el formulario se comporta como si el valor fuera get.”

Atributos comunes

name	Define el nombre de la variable que se envía al servidor.
value	Valor inicial de un campo.
required	Hace que el campo sea obligatorio.
placeholder	Texto temporal que aparece dentro de un campo de entrada.
size	Permite establecer la longitud de los controles de texto.
maxlength y minlength	Permite establecer la longitud máxima/mínima respectivamente.

Campos de entrada

submit	botón para enviar el formulario.
text	entrada de texto.
password	entrada de texto que oculta los caracteres.
email	valida direcciones de correo electrónico.
button	para crear botones (incluyendo los de envío).

Submit

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

Escriba algo:

Enviar

Text

```
<p>Nombre: <input type="text" name="nombre"></p>
```

```
<p>Apellidos:      <input      type="text"      name="apellidos"
size="40"></p>
```

```
<p>Nacionalidad:   <input    type="text"    name="nacionalidad"
size="20"></p>
```

Nombre:

Apellidos:

Nacionalidad:

Password

```
<p>Usuario: <input type="text" name="usuario"
maxlength="6"></p>
```

```
<p>Contraseña: <input type="password" name="password"
minlength="6"></p>
```

Usuario:

Contraseña:

Enviar

Email

```
Correo: <input type="email" name="correo" size="40">
```

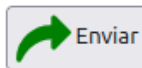
Correo:

Enviar

Button

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

Escriba algo:



Input Button

Los botones se crean mediante la etiqueta `<input>` o mediante la etiqueta `<button>`.

La diferencia entre ellos es que `<input>` sólo puede contener texto, mientras que `<button>` permite incluir elementos html como imágenes.

Elementos adicionales

textarea para entradas de texto largas.

select y option para listas desplegables.

label etiqueta para describir los campos de entrada, que mejora la accesibilidad.

date vlinea específica para fechas (días, meses, años)

textarea

```
<textarea name="texto" rows="3" cols="30"></textarea>
```



select y option

```
<select name="menu">  
  <option>Uno</option>  
  <option>Dos</option>  
  <option>Tres</option>  
</select>
```

Elija una opción:

label


```
<label for="Name">Nombre</label>  
<input type="text" id="Name" name="Name" />  
<label>Apellido<input type="text" id="Name" name="Name" /></label>
```

Nombre

Apellido

date

Fecha:

Fecha: 

Enviar

Multimedia

Multimedia

Un elemento multimedia en HTML puede ser cualquier cosa que puedas ver y oír, desde imágenes y archivos pdf, hasta videos o audios.

Aquí te contamos si las moscas de las flores, indispensables para polinización, son avispas o abejas

NATIONAL GEOGRAPHIC - 27 OCTUBRE, 2024



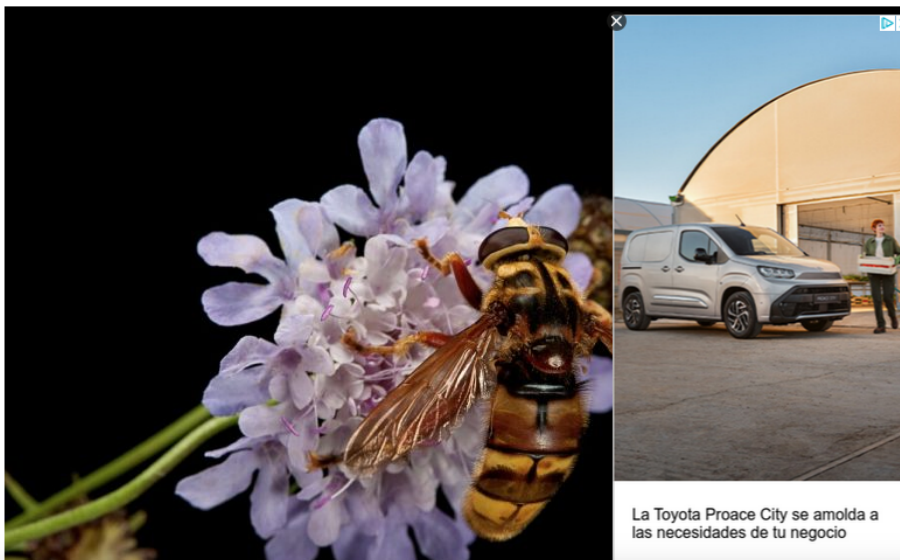
ESCUCHA EL RESUMEN DEL ARTÍCULO

00:00

powerbeans

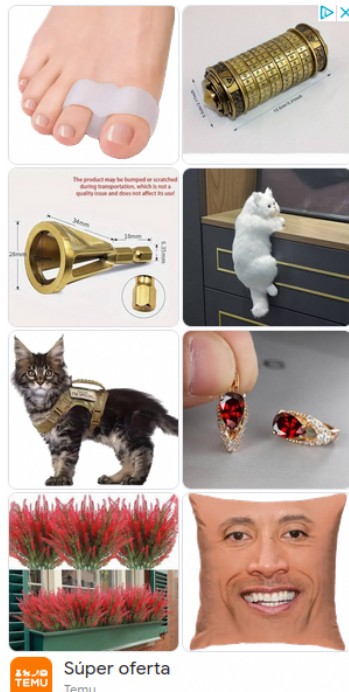
00:48

1x



La Toyota Proace City se amolda a las necesidades de tu negocio

Este sitio utiliza cookies para ayudarnos a mejorar tu experiencia cada vez que lo visites. Al continuar navegando en él, estarás aceptando su uso. Podrás deshabilitarlas accediendo a la configuración de tu navegador.



44-70
TEMU
Super oferta
Temu



Leer Más

Acepto



Article by National Geographic
Published Oct 28, 2024

Aquí te contamos si las moscas de las flores, indispensables para polinización, son avispas o abejas ...

Leer Más

Etiquetas como <iframe> <object> o <canvas> sirven para añadir este tipo de elementos.

Etiqueta `iframe`

La etiqueta `<iframe>`, (inline frame), sirve para añadir a nuestra pagina web contenido de otra distinta u otro archivo Html nuestro, ejemplo: añadir un video de Youtube, o un Mapa de Google Maps.

<> index.html U ●

NAMS > <> index.html > html > body > iframe

```
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      </head>
7      <body style="background-color: aqua;">
8
9          <iframe width="560" height="315"
10             src="https://www.youtube.com/watch?v=ekr2nIex040"
11             title="YouTube video player">
12      </iframe>
13  </body>
14 </html>
```

Etiqueta audio

La etiqueta <audio>, tendría el mismo comportamiento que el anterior pero con extensiones tipo audio de tu propio terminal.

```
<audio src="mi-audio.mp3" type="audio/mpeg"> </audio>
```

Puedes añadir controles para que el usuario tenga control sobre el audio, hacerlo loop o que tenga un determinado tiempo.

Extensiones mas comunes: .mp3, .wav, .ogg



Etiqueta video

La etiqueta <video> seria igual pero con extensiones de tipo video, sirve para añadir videos de tu propio terminal.

```
<video src="mi_video.mp4" type="video/mp4"> </video>
```

Como el anterior, también es posible añadir controles, o un autoplay como en la mayoría de videos publicitarios que conocemos.

Extensiones mas comunes: .mp4, .webm, .ogg



index.html U X

NAMS > index.html > html > body > video

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   </head>
7   <body style="background-color: aqua;">
8
9     <audio src="/video y audio/relaxing-audio-for-yoga-131673.mp3"
10      autoplay controls></audio>
11     <video src="/video y audio/vecteezy_morning-sunrise-reflections-natural-lagoon-krabi-thailand_2375835.mp4"
12      width="600" height="600" muted autoplay controls loop>
13     </video>
14
15   </body>
16 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

No problems have been detected in the workspace.

< master* 0 0 0



Ln 11, Col 106 Spaces: 4 UTF-8 LF HTML Port: 5500 Prettier

Etiquetas object y embed

La etiqueta <object> permite incrustar tanto archivos de texto, como audios y videos, es mas versátil que las otras etiquetas y es el mas recurrido para archivos interactivos.

La etiqueta <embed> es mas especifico para el contenido multimedia en si, y no es obligatorio usar ambas unidas, pero es recomendable hacerlo por problemas de compatibilidad con los diferentes navegadores.

```
<object data="mi-audio.mp3" type="audio/mpeg" width="300"  
height="50">
```

```
<embed src="mi-audio.mp3" type="audio/mpeg" width="300"  
height="50">
```

Si tu enfoque principal es trabajar solo audio y video, usar las etiquetas específicas <audio> y <video> es generalmente la mejor opción por su simplicidad, optimización y experiencia.

La etiqueta <object> es útil cuando necesitas incrustar contenido variado, y usado generalmente para el interactivo.

index.html U X

NAMS > index.html > html

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   </head>
7   <body style="background-color: aqua;">
8
9     <object data="/video y audio/9016.pdf" type="application/pdf" width="300" height="500"> </object>
10    <embed src="/video y audio/9016.pdf" type="application/pdf" width="300" height="500"> </embed>
11  </body>
12 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

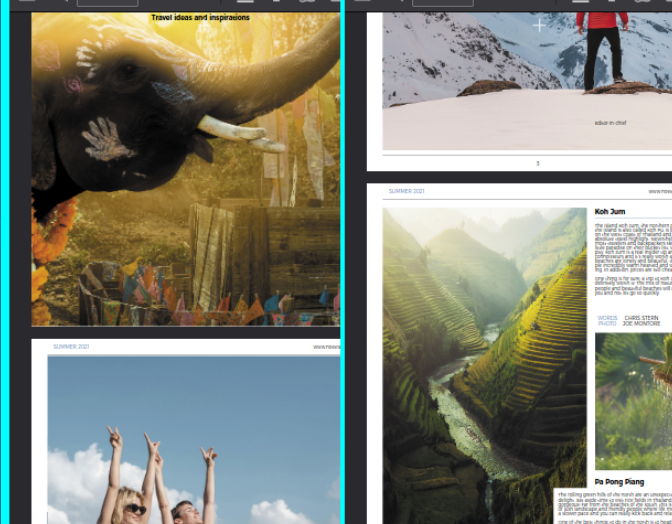
No problems have been detected in the workspace.

master* 0 0 0

127.0.0.1:5500/index.html X +

< > ↻ 127.0.0.1:5500/index.html

1 de 18 — T



Etiqueta canvas

La etiqueta <canvas> se utiliza para dibujar gráficos y animaciones dinámicas mediante Java, es un elemento de contenedor que permite la creación de gráficos 2D y 3D en tiempo real.

```
<canvas id="miCanvas" width="400" height="200" style="border:1px solid black;">
```

Es utilizado para crear gráficos en tiempo real, o para el renderizado de videojuegos creados a través de HTML.



Run >

```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML canvas tag.</canvas>

<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Create gradient
var grd = ctx.createRadialGradient(75,50,5,90,60,100);
grd.addColorStop(0,"red");
grd.addColorStop(1,"white");

// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10,10,150,80);
</script>

</body>
</html>
```



index.html U X

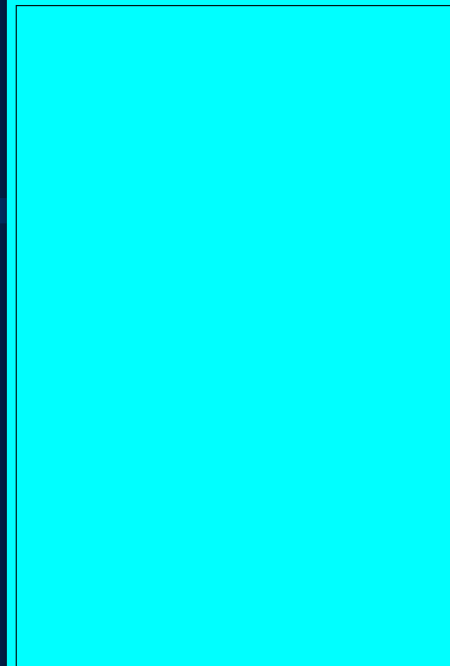
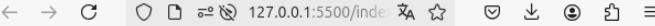
NAMS > index.html > html

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   </head>
7   <body style="background-color: aqua;">
8     <canvas id="miCanvas" width="400" height="600" style="border:1px solid black;"></canvas>
9   </body>
10 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

No problems have been detected in the workspace.

127.0.0.1:5500/index.html X +



Imágenes

Etiqueta `img`

Introducción de imágenes. No tiene par de cierre.

Atributos obligatorios: `src` y `alt` / **Atributos recomendables:** `height` y `width`

```

```



Etiqueta figure

Con <figure> y <figcaption> se consigue agrupar figura y su leyenda también se puede aplicar a video, audio.

```
<figure>
  
  <figcaption>Preciosa imagen de una atardecer vista desde la playa</figcaption>
</figure>
```



Preciosa imagen de una atardecer vista desde la playa

Etiquetas picture y source

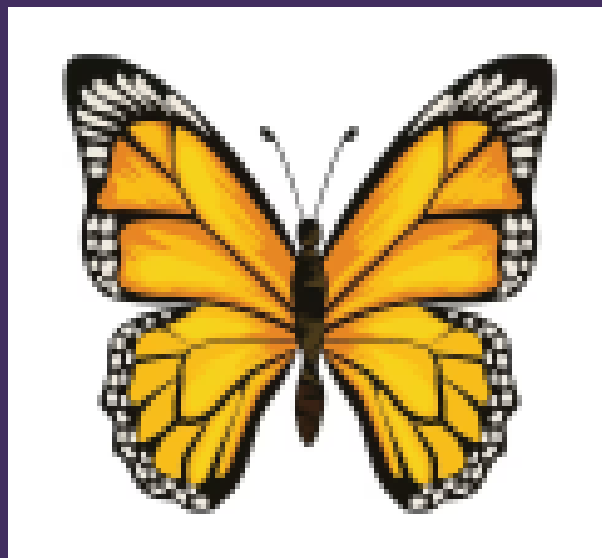
<picture> Se utilizará como un contenedor donde vamos a indicar al navegador que queremos mostrar una imagen. Luego, en su interior, podemos establecer varias etiquetas <source> donde vamos a proporcionarle posibles opciones, de las cuáles tendrá que decidir cuál utilizar.

```
<picture>
  <source srcset="image.avif">
  <source srcset="image.webp">
  
</picture>
```

Responsive

Otra ventaja interesante es que con `<picture>` podemos crear imágenes responsive que cambien dependiendo de características de las media queries (CSS). Por ejemplo, utilicemos `min-width` (tamaño mínimo de ancho de la pantalla) en el siguiente ejemplo:

```
<picture>
  <source media="(width >= 900px)"
    srcset="logo-large.png,
            logo-large-hd.png 2x,
            logo-large-fhd.png 3x">
  <source media="(width >= 600px) and (width < 900px)"
    srcset="logo-medium.png,
            logo-medium-hd.png 2x,
            logo-medium-fhd.png 3x">
  <source media="(width < 600px)"
    srcset="logo-small.png,
            logo-small-hd.png 2x,
            logo-small-fhd.png 3x">
  
</picture>
```



Etiqueta svg

La etiqueta `<svg>` es el elemento padre de cualquier imagen vectorial svg, ya que todos los elementos deben estar en su interior.

Sus atributos mas importantes son:

El atributo **xmlns** namespace indica una URL donde se encuentra información sobre la especificación que se va a utilizar de SVG. Lo más habitual, suele ser indicar la URL

```
<svg xmlns="imagenes/mariposa-amarilla_.avif">  
  <path d="M0,0 V50 L50,0 Z" fill="indigo" />  
</svg>
```

```
<svg xmlns="imagenes/mariposa-amarilla_.avif" style="background: ■ #150453">  
  <path d="M0,0 V50 L50,0 Z" fill="indigo" />  
</svg>
```

Viewbox

El atributo viewBox se utiliza para determinar las dimensiones del lienzo de la imagen SVG. Observa el ejemplo anterior, donde he añadido el atributo style para darle estilos CSS de color de fondo al elemento <svg> y determinar visualmente cuál es el espacio que está ocupando por defecto:



Gracias

//Nahia //Allan //Moisés //Silvia