

APUNTES HTML Y CSS

Mara Arpa Recio

marparecio@educa.jcyl.es

Índice

INTRODUCCIÓN.....	2
DOCUMENTO HTML 5.....	2
ESTRUCTURA DEL ARCHIVO.....	2
ETIQUETAS PARA LA CABECERA.....	3
ETIQUETAS PARA EL CUERPO.....	3
Divisiones	3
Formato.....	3
Listas.....	4
Tabla.....	6
Imágenes.....	7
Enlaces	7
OTRAS ETIQUETAS, HTML5	8
Audio	8
Video	9
Embed	9
Iframe.....	9
Canvas	10
Otras etiquetas.....	11
FORMATO CON CSS (estilo)	12
FORMAS DE DAR ESTILO	12
PSEUDO CLASES	14
FONDOS.....	14
TEXTO	16
CAJAS.....	17
LISTAS.....	19
TABLAS	20
IMÁGENES	21
ENLACES	21
CURSOR.....	21
ANIMACIONES.....	22
EFFECTOS	25
CÓDIGO DE COLORES:.....	26
FUENTES CONSULTADAS.....	32

INTRODUCCIÓN

Para facilitar la búsqueda, las **etiquetas** aparecerán en azul, los **atributos** en rojo y los posibles **valores** en morado.

Crear carpeta para guardar todos los archivos de la página web. *PaginaWeb_nombre*

Primer paso: crear un nuevo documento en notepad++ (o bloc de notas) y guardarlo como index.html, eligiendo como tipo `hypertextmarkuplanguage`(o “todos los archivos” en bloc de notas) para que lo guarde como una página web y no como texto.

Todos los archivos tienen que estar guardados en la misma carpeta, si se crean carpetas dentro para ordenar, por ejemplo fotos, se debe especificar en las rutas.

Los documentos que creamos deben estar documentados, tanto para que los comprendan terceras personas, como para ayudarnos a nosotros mismos a organizar el documento. Para incluir comentarios en HTML tenemos que usar la etiqueta `<!-- -->` dentro de la cuál escribiremos el texto.

```
<!--comentario, este texto no se muestra en la página ni influye en el archivo-->
```

DOCUMENTO HTML 5

Para declarar que el documento está creado bajo HTML5, para ellos se usa la etiqueta `<!DOCTYPE html>`, en la primera línea del documento.

Para que se reconozcan la mayoría de los caracteres usado, y no aparezcan símbolos raros en tildes o en la letra ñ, usaremos la etiqueta. `<meta charset="UTF-8">`. La codificación UTF-8 también es la que mejor reconocen los teléfonos.

ESTRUCTURA DEL ARCHIVO.

HTML es un lenguaje que se basa en etiquetas, las etiquetas se identifican por los símbolos de `<>` donde se abre la etiqueta y por `</>` donde se cierra. Normalmente entre las etiquetas de apertura y cierre se incluye el contenido.

<HTML> (Etiqueta que indica que estamos creando una página web)

<HEAD>(Etiqueta de apertura de la cabecera).

En esta parte se incluye información sobre el documento como título, autor, enlaces, estilo... lo único visible en la página es el título, que aparecerá en la pestaña del navegador.

</HEAD>(Etiqueta de cierre de la cabecera)

<BODY>(Etiqueta de apertura del cuerpo)

Aquí va el contenido de la página que será lo que se vea en el navegador.

</BODY>(Etiqueta de cierre del cuerpo)

</HTML>(Etiqueta de cierre del documento)

Para facilitar que el navegador comprenda las partes de la página web podemos usar, dentro de `<body>`, etiquetas para diferenciar encabezado, parte principal y pie de página. Estas etiquetas no afectan a la apariencia de la página, pero facilitan la labor de las herramientas de accesibilidad, como el dictado.

<header> **</header>** suele contener logo, título, descripción o información sobre el sitio y los botones de navegación. El menú de navegación, se puede delimitar por la etiqueta **<nav>****</nav>** que indicará al navegador que es una secuencia de enlaces

`<main></main>` incluye el contenido principal

`<footer></footer>` En esta parte suele ir la información sobre protección de datos, contacto, diseño de la web...

ETIQUETAS PARA LA CABECERA

La cabecera contiene información sobre el documento HTML y su configuración, como título, enlaces, script o metadatos. La mayoría de la información escrita en este apartado no se muestra en la página web, una excepción en el título.

`<title>` Aquí se escribe el título de la página que aparecerá en la pestaña del navegador `</title>`

Resultado:



`<link href="style.css" rel="stylesheet" type="text/css">` es la etiqueta que enlaza el documento style.css con el HTML para poder cambiar formatos de texto, fondo, imágenes... (tamaño, fuente, color...)

En el encabezado podemos incluir la etiqueta `<script>`, un fragmento o enlace a un archivo de código fuente, el lenguaje más usado es Java, pero también podemos usar Processing.

```
<script src="javascript.js"></script>
```

```
<script src="processing-1.1.0.min.js"></script>
```

El atributo esencial de esta etiqueta es `src`, la ruta del archivo. (Aunque se podría incluir el código directamente dentro de la etiqueta, es preferible usar un archivo externo).

En las versiones anteriores a Html 5, también era necesario incluir el atributo `Type`, donde se indicaba el lenguaje del script. (`text/javascript`, `application/javascript` ...)

ETIQUETAS PARA EL CUERPO

Divisiones

Para poder dividir la página en partes podemos usar la etiqueta `<div></div>` esta etiqueta debe llevar el atributo `id`, de esta forma podremos identificar las distintas zonas de nuestra página. Para lograr un diseño atractivo, jugaremos con las cajas en CSS (posicionamiento, tamaño, márgenes, fondo y demás de cada división).

Formato

El formato de una página web se puede hacer de tres formas distintas: dentro de cada etiqueta, usando código CSS en la etiqueta `Style` dentro de la cabecera, o con un archivo CSS independiente enlazado al archivo de html. La opción más práctica, y la que vamos a usar, es la 3ª. Por ello no vamos a ver todas las etiquetas y atributos que nos permiten dar formato en Html, tan solo vamos a ver algunos de ellos.

El texto que se escriba o se copie en un documento html no tendrá formato, aparecerá como un arial 11 de color negro, sin saltos de línea.

- Saltos de línea

`
` hacer un salto de línea.

`<p>` Limita un párrafo, con un doble salto de línea después `</p>`

- Alineación

`<center>`centra en la pantalla el texto, tabla o imagen incluido`</center>`

- Formato del texto:

`` pone en negrita el texto que está dentro de estas etiquetas``

`<i>` pone en cursiva `</i>`

`<u>` subraya`</u>`

`<mark>` resalta el texto `</mark>`

Texto

Texto escrito en línea distinta

`
` Texto después de hacer un salto de línea.

`<p>`Texto dentro de un parrafo`</p>`

`<center>`texto centrado`</center>`

`
`

Texto normal, `` texto en negrita, `` `<I>` texto en cursiva, `</I>` `<U>` texto subrayado`</U>` y

`<mark>` texto resaltado `</mark>`

Texto Texto escrito en línea distinta

Texto después de hacer un salto de línea.

Texto dentro de un parrafo

texto centrado

Texto normal, **texto en negrita**, *texto en cursiva*, texto subrayado y **texto resaltado**

Caracter	Se escribe	Caracter	Se escribe
Espacio en blanco	<code>&nbsp;</code>	±	<code>&plusmn;</code>
¡	<code>&iexcl;</code>	´	<code>&acute;</code>
¶	<code>&para;</code>	¨	<code>&uml;</code>
©	<code>&copy;</code>	»	<code>&raquo;</code>
«	<code>&laquo;</code>	®	<code>&reg;</code>
¿	<code>&iquest;</code>	á	<code>&aacute;</code>
é	<code>&eacute;</code>	í	<code>&iacute;</code>
ó	<code>&oacute;</code>	ú	<code>&uacute;</code>
ü	<code>&uuml;</code>	<	<code>&lt;</code>
>	<code>&gt;</code>	ñ	<code>&ntilde;</code>
"	<code>&quot;</code>	&	<code>&amp;</code>

Listas

En html podemos incluir listas en las que los elementos pueden aparecer desordenados, precedidos por un símbolo, u ordenadas, precedidos por número o letras.

Listas desordenadas (con puntos, cuadrados o guiones)

La etiqueta es ``, y dentro de ella usaremos `` para cada elemento. Podemos usar el atributo **type** para definir el símbolo o modificarlo en CSS.

``Empieza listado

`<li type="circle">`texto de cada punto de la lista, el símbolo es un círculo.``

`<li type="square">`texto de cada punto de la lista, el símbolo es un cuadrado.``

`<li type="disc">`texto de cada punto de la lista, el símbolo es un punto, si no se pone estilo es el que aparece por defecto.``

``Termina el listado

Para que todos los elementos de la lista tengan el mismo símbolo, incluimos el argumento *type* dentro de la etiqueta ``

Lista no ordenada por defecto

```
<ul>
  <li> elemento de la lista</li>
  <li> elemento de la lista</li>
</ul>
```

Lista no ordenada con cuadrados

```
<ul type="square">
  <li> elemento de la lista</li>
  <li> elemento de la lista</li>
</ul>
```

Lista no ordenada con círculos

```
<ul type="circle">
  <li> elemento de la lista.</li>
  <li> elemento de la lista</li>
</ul>
```

Lista no ordenada con discos

```
<ul type="disc">
  <li> elemento de la lista.</li>
  <li> elemento de la lista</li>
</ul>
```

Lista ordenada con distintos símbolos en cada punto

```
<ul>
  <li type="square"> elemento de la lista</li>
  <li type="circle"> elemento de la lista.</li>
  <li type="disc"> elemento de la lista.</li>
</ul>
```

Lista no ordenada por defecto

- elemento de la lista
- elemento de la lista

Lista no ordenada con cuadrados

- elemento de la lista
- elemento de la lista

Lista no ordenada con círculos

- elemento de la lista.
- elemento de la lista

Lista no ordenada con discos

- elemento de la lista.
- elemento de la lista

Lista ordenada con distintos símbolos en cada punto

- elemento de la lista
- elemento de la lista.
- elemento de la lista.

Listas ordenadas

Para las listas ordenadas usamos la etiqueta `` e igual que en las no ordenadas, `` para cada elemento.

`` empieza el listado

`` etiquetas para cada elemento de la lista ``

`` termina el listado

Puedo elegir el tipo de ordenación entre numeración decimal, que es la lista ordenada por defecto, alfabética o romana usando el atributo **type**. También puedo forzar a un elemento para que tenga un determinado valor, en ese caso, los siguientes elementos seguirán la numeración desde el valor forzado, con el atributo **value**.

Lista ordenada por defecto

```
<ol>
  <li>texto de cada punto de la lista .</li>
  <li>texto de cada punto de la lista .</li>
</ol>
```

Lista ordenada decimal

```
<ol type="1"> <!-- argumento para listas ordenadas tipo decimal-->
  <li>texto de cada punto de la lista .</li>
  <li>texto de cada punto de la lista .</li>
</ol>
```

Lista ordenada alfabética

```
<ol type="a"> <!-- argumento para listas ordenadas tipo alfabético-->
  <li>texto de cada punto de la lista .</li>
  <li>texto de cada punto de la lista .</li>
</ol> Termina el listado
```

Lista ordenada romanos

```
<ol type="i"> <!-- argumento para listas ordenadas tipo numeración romana-->
  <li>texto de cada punto de la lista .</li>
  <li>texto de cada punto de la lista .</li>
</ol>
```

Lista ordenada empezando en 5

```
<ol type="i"> <!-- argumento para listas ordenadas tipo numeración romana-->
  <li value="5">texto de cada punto de la lista .</li>
  <li>texto de cada punto de la lista .</li>
</ol>
```

Lista ordenada por defecto

1. texto de cada punto de la lista .
2. texto de cada punto de la lista .

Lista ordenada decimal

1. texto de cada punto de la lista .
2. texto de cada punto de la lista .

Lista ordenada alfabética

- a. texto de cada punto de la lista .
- b. texto de cada punto de la lista .

Lista ordenada romanos

- i. texto de cada punto de la lista .
- ii. texto de cada punto de la lista .

Lista ordenada empezando en 5

- v. texto de cada punto de la lista .
- vi. texto de cada punto de la lista .

Tabla

Para crear tablas usamos la etiqueta `<table>`, dentro de la tabla primero hay que crear filas, y dentro de estas columnas. La etiqueta para las filas es `<tr>` y para las columnas `<td>`.

Podemos darle título a la tabla usando la etiqueta `<caption></caption>`

```
<table class="tabla1">
  <caption>Titulo tabla</caption>
  <tr>
    <td>celda 1</td>
    <td>celda 2</td>
  </tr>
  <tr>
    <td>celda 3</td>
    <td>celda 4</td>
  </tr>
</table>
```

Título tabla celda 1 celda 2 celda 3 celda 4
--

Para que se vean las líneas que delimitan las celdas podemos usar el atributo `border`, con el grosor de la línea, en la etiqueta `table` o darle formato a las celdas con CSS.

Dentro de las columnas creadas podemos usar los atributos `rowspan` para combinar filas, y `colspan` para combinar columnas.

```
<table>
  <tr> <!-- abre fila-->
    <td> celda 1 </td> <!-- crea 1ª columna-->
    <td> celda 2 </td> <!-- crea 2ª columna-->
  </tr> <!-- cierra fila-->
  <tr> <!-- abre fila-->
    <td> celda 3 </td> <!-- crea 1ª columna-->
    <td> celda 4 </td> <!-- crea 2ª columna-->
  </tr> <!-- cierra fila-->
</table>
<br/>
Tabla con líneas
<table border="1"> <!-- se ven las líneas de la tabla-->
  <tr> <!-- abre fila-->
    <td> celda 1 </td> <!-- crea 1ª columna-->
    <td> celda 2 </td> <!-- crea 2ª columna-->
  </tr> <!-- cierra fila-->
  <tr> <!-- abre fila-->
    <td> celda 3 </td> <!-- crea 1ª columna-->
    <td> celda 4 </td> <!-- crea 2ª columna-->
  </tr> <!-- cierra fila-->
</table>
<br/>
Tabla con celdas combinadas
<table border="1">
  <tr>
    <td colspan=3>celda combinando 3 columnas</td><!-- combina tres celdas de columnas distintas en una.-->
  </tr>
  <tr>
    <td rowspan=2>celda combinando 2 </td><!--combina dos celdas de filas distintas en una.-->
    <td>celda normal</td>
    <td>celda normal</td>
  </tr>
  <tr>
    <td>celda normal</td>
    <td>celda normal</td>
  </tr>
</table>
```

Tabla por defecto
celda 1 celda 2
celda 3 celda 4

Tabla con líneas

celda 1	celda 2
celda 3	celda 4

Tabla con celdas combinadas

celda combinando 3 columnas		
celda combinando 2	celda normal	celda normal
	celda normal	celda normal

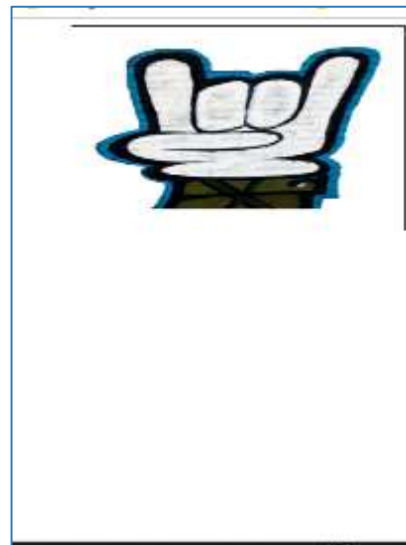
Imágenes

Para incluir imágenes en un documento HTML, usamos la etiqueta `` dentro de la cuál necesitaremos como atributo esencial la ruta de la imagen a mostrar, `src`. Además podemos incluir otros atributos como los siguientes, aunque es recomendable dar el formato usando con CSS.

```

```

- **Src**: ruta y nombre de la imagen. Si la imagen está en la misma carpeta que el documento html, no hace falta incluir la ruta, si está en otra carpeta, o en una sitio web, tendremos que incluir la ruta completa.
- **alt**: texto si no se ve la imagen,
- **width**: anchura en pixeles (también puede poderse por porcentaje)
- **height**: altura
- **border**: pone borde a la imagen.
- **align**: utilizando los atributos "left" para alinearla a la izquierda, "right" para alinearla a la derecha, "top" para alinearla arriba, "bottom" para alinearla abajo y "middle" para alinearla al centro.



```

```

Enlaces

Para crear enlaces tanto internos (en el propio documento) como externos, la etiqueta a usar es `<a>`, el atributo esencial de esta etiqueta es `href`, que indica a que estamos referenciando con el enlace. Entre la etiqueta de apertura y cierra tenemos que escribir el texto que servirá de enlace al pinchar sobre él, o incluir la imagen que actúe como un botón, haciendo que vayamos a otra página, documento, o lugar de la misma página al pinchar sobre ella.

Igual que con las imágenes, debemos incluir la ruta completa, en el caso de enlazar a documentos, si están guardados en la misma carpeta que el archivo html, tan solo necesitaremos el nombre.

Podemos crear enlaces externos, a otra página o documento, e internos, a un apartado de la propia página.

`texto sobre el que pinchar` es un texto que sirve de enlace a google.

`texto sobre el que pinchar` es un texto que sirve de enlace a google y que se abre en una pestaña nueva.

`texto sobre el que pinchar` es un texto que sirve para descargar un documento de word.

`texto sobre el que pinchar` es un texto que sirve para otro archivo de la página web.

`ver punto 2` el texto sirve de enlace para que se muestre el párrafo donde está el punto 2.

En ese párrafo necesitamos poner la identificación llamada con #

`<p id="punto2">El punto 2 del tema es</p>`


```

<p id="inicio"> Este es el inicio de la página.</p>
<p>
<a href="https://www.google.es"> enlace para cargar google en esta pestaña</a>
</p>
<p>
<a href="https://www.google.es" target="_blank"> enlace para cargar google en una nueva pestaña</a>
</p>
<p>
<a href="trabajo.docx ">enlace para abrir un documento de word</a>
</p>
<p>
<a href="pagina.html">enlace para cargar otro documento html en esta pestaña</a>
</p>
<p>
<a href="#inicio"> enlace para ir al inicio de esta  página </a>
</p>

```

OTRAS ETIQUETAS, HTML5

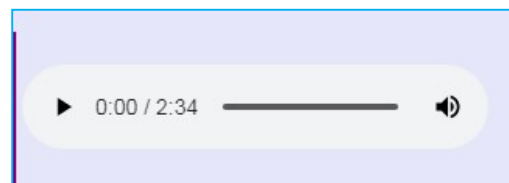
En este apartado vamos a incluir otras etiquetas más avanzadas, incluidas las propias de Html 5.

Audio

<audio></audio> Esta etiqueta permite incluir un archivo de audio en la página web para que se reproduzca. El atributo esencial es **src**, pero podemos incluir otros atributos que nos permiten configurar la reproducción .

- **src** Ruta del audio
- **controls** Muestra los controles de reproducción.

```
<audio src="audio.mp3" controls >
```



Hay que tener en cuenta los formatos de audio, la calidad que tienen, si son recomendables para reproducir en navegadores y los códec que necesitan.

Formato	Codec utilizado	Características
MP3	MPEG Layer-3	Buena calidad. Buen soporte
OGG	Ogg Vorbis	Buena calidad. Alternativa libre a MP3. Soporte medio
AAC	Advanced Audio Coding	Mejora el MP3. Usado como audio en MP4. Soporte medio
Opus	Opus	Buena calidad. Alternativa libre a MP3. Soporte malo
FLAC	FLAC Audio Lossless	Compresión sin pérdidas. Alto tamaño
WAV	Wave sound	Formato de Microsoft. Está soportado. Muy pesado

Para conseguir mayor compatibilidad con el navegador, o incluir opciones adicionales, como los subtítulos, podemos usar etiquetas como **<source>** o **<track>** dentro ella,

La etiqueta **<source>** para incluir varios formatos de un mismo audio, el navegador reproducirá el primero que sea compatible. Tiene como atributos **src** con la ruta del audio y **type** que especificará los códec que usa.

<track> permite incorporar subtítulos. Sus atributos son: **src**, (contiene la ruta), **srclang** (código idioma), **label** (descripción del idioma), **kind** (tipos de subtítulo, puede tener los valores: **subtitles**: subtítulo sobre el reproductor, **metadata** son datos técnicos, no se ven, **chapters**: título de los capítulos, **captions**: transcripción de audio, incluye música, efectos.. **descriptions**: Descripción textual del contenido del vídeo)

```

<audio controls >
  <source src="audio.opus">
  <source src="audio.ogg">
  <source src="audio.mp3">
</audio>

```

Video

`<video>` `</video>` Esta etiqueta permite incrustar un video, el atributo esencial es la ruta `src`, pero para que se pueda interactuar con el video deberemos incluir otros atributos como `Controls`. Funciona como la etiqueta `audio`, aunque a mayores se puede configurar el tamaño del video, `width` ancho del video y `height` alto del video.

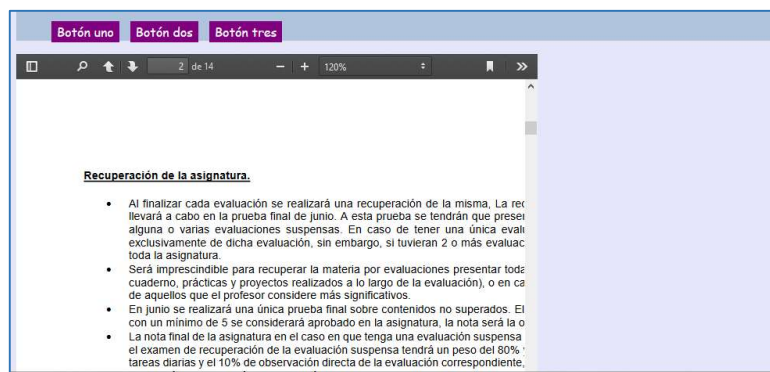
```
<video src="video.mp4" width="640" height="360" controls ></video>
```

También podemos usar la etiqueta `video`, para que contenga las etiquetas `source` y `track`, igual que en el `audio`.

Embed

La etiqueta `<embed>` `</embed>` permite integrar contenido externo, como aplicaciones o archivos. Los atributos que admite son **`src`**, **`type`**, **`width`**, **`height`**. Dentro de la ruta del archivo podemos incluir parámetros después de **`#`**(para usar varios se usa **`&`** entre ellos) como: **`page=número`** (muestra la página del documento indicada) o **`zoom=número`** (100 representará un zoom del 100%) entre otros.

```
<embed src="criterios.pdf#page=2&zoom=120" type="application/pdf" width="50%" height="600px" />
```

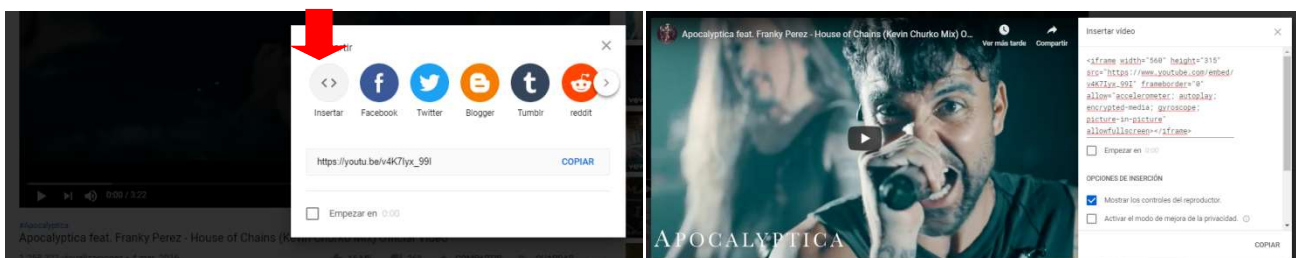


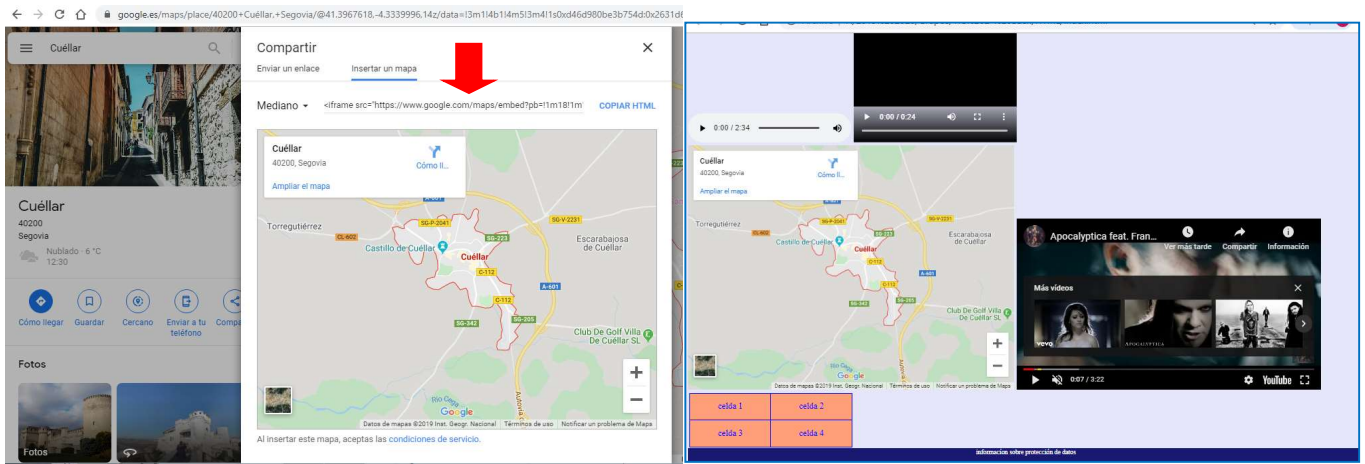
Iframe

Para poder incrustar contenido de otro sitio web, podemos usar la etiqueta `<iframe>` como atributos esenciales tenemos **`src`**, **`width`** y **`height`**.

Además, podemos incluir otros atributos como **`importance`** con valores **`auto`**, **`high`** y **`low`**, que le indican al navegador la prioridad que tiene el contenido del `iframe` dentro de la página para poder gestionar los recursos., o **`allowfullscreen`** con el valor **`true`**, si se quiere permitir que se ponga a pantalla completa.

Para incluir contenido de algunas páginas como google maps, o youtube, debemos usar la opción insertar dentro de compartir, de esta forma nos mostrará la etiqueta `iframe` con la ruta necesaria y una serie de atributos por defecto que se podrán modificar una vez lo hayamos copiado al archivo `html`.





Canvas

La etiqueta `<canvas>` permite incluir gráficos animados, como pueden ser los banner de publicidad, las simulaciones o imágenes interactivas, mediante scripts (como javascript o processing). Está etiqueta es la que va a permitir que se pueda seguir usando juegos online cuando flash termine de desaparecer.

Para poder asociar processing con la etiqueta canvas, primero se debe usar la etiqueta script en la cabecera.

Debemos usar una primera etiqueta para enlazar el archivo `processing.js`, que debe estar en la carpeta del archivo html, que permitirá que se ejecute el código en el navegador. (se puede descargar de la página <http://processingjs.org/download/>)

Después crearemos otro script para el canva que queremos mostrar, en este usaremos el atributo `data-processing-target` para indicar obajetivo, identificador de la etiqueta Canva, que va a contenerlo.

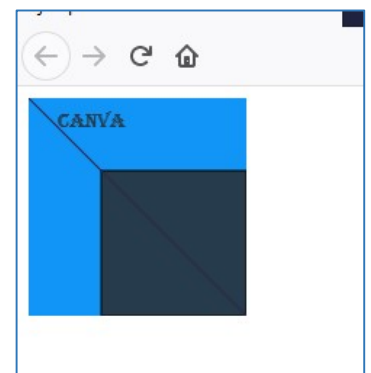
```
<script src="processing.js"></script>
<script type="text/processing" data-processing-target="mycanvas" >
```

Dentro del segundo script copiaremos el código Processing.

```
<script src="processing.js"></script>
<script type="text/processing" data-processing-target="mycanvas" >
  void setup()
  {
    size(150,150);
    background(17,149,247);
    fill(#263B4B);
    quad(50,50,50,150,150,150,150,50);
    stroke(#2F264B);
    line(0,0,150,150);
    noLoop();
    PFont fontA = createFont("algerian",1);
    textFont(fontA, 14);
  }
  void draw(){
    text("Canva",20,20);
  }
</script>
```

En el cuerpo del documento HTML, incluiremos la etiqueta `<canvas>` con el identificador que hemos indicado como objetivo en el script.

```
<canvas id="mycanvas" ></canvas>
```



Otras etiquetas

<article>	Define contenido autónomo que podría existir independientemente del resto del contenido.
<aside>	Define algunos contenidos vagamente relacionados con el resto del contenido de la página. Si se quita el resto de contenido sigue teniendo sentido.
<bdi>	Representa un texto que debe ser aislado de sus alrededores para el formateado bidireccional del texto. Permite incrustar un fragmento de texto con una direccionalidad diferente o desconocida.
<command>	Representa un comando que un usuario puede invocar.
<details>	Representa un widget desde el que un usuario puede obtener información o controles adicionales.
<figure>	Representa una figura ilustrada como parte del documento.
<figcaption>	Representa la leyenda de una figura.
<form>	Crear un formulario para enviar información a través de la página. Llevará elementos. <input> de distintos tipos, como áreas de texto, selectores, desplegable, textos de tipo búsqueda, url, teléfono, correo, fecha, color... necesitan ir acompañados de programación (php, java..) para enviar la información.
<math>	Define una fórmula matemática.
<menu>	Representa una lista de comandos
<meter>	Representa la medida escalar (o el valor fraccionario) dentro de un rango conocido.
<output>	Representa el resultado de un cálculo.
<progress>	Representa el progreso de finalización de una tarea.
<section>	Define una sección en un documento.
<summary>	Representa un resumen, título o leyenda para un elemento <details> dado.
<svg>	Define una imagen vectorial embebida.
<wbr>	Representa una oportunidad de salto de línea, es decir, un punto sugerido de envoltura donde el texto de múltiples líneas puede ser dividido para mejorar su legibilidad.

FORMATO CON CSS (estilo)

Para facilitar la búsqueda, usaremos el azul oscuro para las **propiedades** y el morado para los posibles **valores**.

FORMAS DE DAR ESTILO

Para dar distintos formatos al texto, tablas, fondos y demás componentes de un documento HTML usamos archivos externos CSS, son archivos que contienen todos los estilos que van a usarse en la página web.

En un archivo nuevo guardado como style.css (tipo cascadestylesheetfile) creamos el código para dar estilo.

Para crear estilos en CSS, usaremos selectores, como * para todos los elementos, el nombre de las etiquetas de Html o selectores de estilo propios (incluyendo un punto antes del nombre del estilo), selector de ID (incluyendo # antes del id del elemento). Si usamos las etiquetas de html, el formato afectará a todo el contenido que esté dentro de todas las etiquetas nombradas. Por ejemplo, si en el estilo body indico que el texto será de color azul. Todo el texto de la página será azul.

```
body{  
    color:blue;  
}
```

```
<!DOCTYPE html>  
<HTML>  
  <HEAD>  
    <title>TITULO</title>  
    <link href="estilo.css" rel="stylesheet" type="text/css">  
  </HEAD>  
  <BODY>  
    <p> MI EMPRESA</p>  
    <p> Nombre de la empresa <br/>  
    Dirección de la empresa  
  </p>  
    <table>  
      <tr>  
        <td>celda 1</td>  
        <td>celda 2</td>  
      </tr>  
      <tr>  
        <td>celda 3</td>  
        <td>celda 4</td>  
      </tr>  
    </table>  
    menú  
    <nav>  
      <ul>  
        <li>Botón uno</li>  
        <li>Botón dos</li>  
        <li>Botón tres</li>  
      </ul>  
    </nav>  
  </BODY>  
</HTML>
```

MI EMPRESA

Nombre de la empresa
Dirección de la empresa

celda 1 celda 2
celda 3 celda 4
menú

- Botón uno
- Botón dos
- Botón tres

Si uso la etiqueta p como un estilo, y le indico fuente, color, alineación y decoración al texto, veremos cómo solo afecta al texto del documento HTML contenido en las etiquetas <p>

```
body{  
    color:blue;  
}  
p{  
    font-family:"Tahona";  
    font-size:20px;  
    color:purple;  
    text-align:justify;  
}
```

MI EMPRESA

Nombre de la empresa
Dirección de la empresa

celda 1 celda 2
celda 3 celda 4
menú

- Botón uno
- Botón dos
- Botón tres

Si quiero que el formato solo afecte a unos párrafos concretos, crearé un estilo propio (.nombre_estilo{}) y en la etiqueta html incluiremos el atributo **class** con el estilo elegido. En vez de usar el atributo class, también puedo usar el atributo **id**, en este caso en el archivo css en vez de preceder el nombre del estilo por un punto, usaremos #.

```

<!DOCTYPE html>
<HTML>
  <HEAD>
    <title>TITULO</title>
    <link href="estilo.css" rel="stylesheet" type="text/css">
  </HEAD>
  <BODY>
    <p class="titulo"> MI EMPRESA</P>
    <p class="empresa"> Nombre de la empresa <br/>
    Dirección de la empresa
  </p>
    <table>
      <tr>
        <td>celda 1</td>
        <td>celda 2</td>
      </tr>
      <tr>
        <td>celda 3</td>
        &td>celda 4</td>
      </tr>
    </table>
    menú
    <nav >
      <ul>
        <li>Botón uno</li>
        <li>Botón dos</li>
        <li>Botón tres</li>
      </ul>
    </nav>
  </BODY>
</HTML>

```

```

body{
  color:blue;
}
.titulo{
  font-family:"Algerian";
  font-size:72px;
  color:purple;
  text-align:center;
}
.empresa{
  font-family:"Tahona";
  font-size:20px;
  color:purple;
  text-align:justify;
}

```

Puedo usar selectores descendentes, de forma que el estilo solo afecta a un elemento incluido en otro, para ello escribiremos el estilo del elemento y seguido el descendiente.

```

#menuhorizontal ul li {
  display: inline;
  padding: 2px 7px; /*Separar el texto*/
  margin:5px; /* Separar los botones */
  background-color:#800080;
  color:white;
  font-family:"comic Sans MS";
  font-size:"20px";
}

```

```

<nav id="menuhorizontal">
  <ul>
    <li>Botón uno</li>
    <li>Botón dos</li>
    <li>Botón tres</li>
  </ul>
</nav>

```

En el ejemplo el estilo afectará a cada elemento de la lista del menú de navegación identificado como menuhorizontal.

MI EMPRESA

Nombre de la empresa
Dirección de la empresa

celda 1 celda 2
celda 3 celda 4
menú

- Botón uno
- Botón dos
- Botón tres

PSEUDO CLASES

Representan estados, se incluyen en el archivo CSS después de los estilos creados, indican que cambiará el estilo de un elemento en función del estado en el que se encuentre.

Estados más comunes:

- **:active**, indica que se está pulsando el botón del ratón sobre el elemento.
- **:focus**, se ha seleccionado el elemento.
- **:hover**, el ratón está encima del elemento.

Estados exclusivos de enlaces:

- **:link**, indica que el enlace no ha sido visitado.
- **:visited**, indica que el enlace ha sido visitado.

Cuando se configuran varias pseudo clases se debe seguir el siguiente orden, LVHA (link, visited, hover, active).

FONDOS

Podemos dar formato al fondo usando la propiedad **background**

- Dar **color** al fondo de la página, usaremos la propiedad específica **background-color** seguida del nombre del color (Green) o con el código hexadecimal el color (#66CDAA).

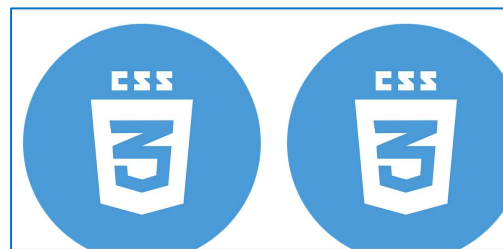
```
body{  
    background-color:#66CDAA;  
}  
  
header{  
    background-color:darkcyan;  
    min-height:200px;  
}
```



**la etiqueta header contiene la propiedad min-height, para que ocupe un espacio aunque no tenga contenido en el documento html.*

- **Imagen** de fondo. Para ellos usaremos la propiedad **background-image: url("imagen.jpg");**

```
body{  
    background-image:url("css.png");  
}
```



Se verá la parte de la imagen que permita el tamaño tanto del contenedor como de la imagen, al no ocupar todo el espacio horizontal la ha repetido.

Con **background-repeat** podremos indicar si queremos que la imagen se repita (**repeat**), se repita en horizontal(**repeat-x**), en vertical (**repeat-y**) o no se repita (**no-repeat**).

También contamos con las propiedades **background-attachment** para indicar la imagen se va a mover al movernos por la página (**scroll**), o si queda fija (**fixed**), y **background-position** para la posición de la imagen, esta posición se puede indicar por los valores de x e y (**0%,0%**) o por la palabra que indica la alineación de la imagen (**left, right, center, top, bottom**).

```
body{
  background-color:darkcyan;
  background-image:url("css.png");
  background-repeat: no-repeat;
  background-attachment:fixed;
  background-position: right;
}
```



Si queremos incluir varias de estas propiedades, como en el ejemplo anterior, podemos incluirlas seguidas tras la propiedad genérica background.

```
body{
  background:darkcyan url("css.png")no-repeat fixed right;
}
```

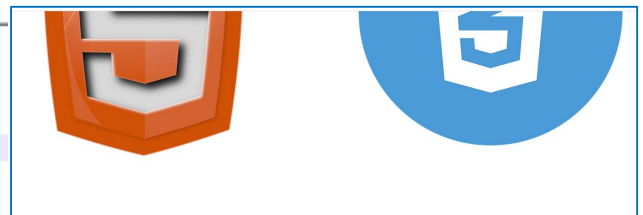
La propiedad de CSS3 **background-size** modifica el tamaño de la imagen, para que la imagen no se escale (**auto**), se escale al alto de la página (**contain**), se escale al ancho (**cover**), o se escale al tamaño que le indiquemos con porcentajes o pixeles (**50%x50%**).

```
body{
  background:darkcyan url("css.png")no-repeat fixed right;
  background-size:cover;
}
```



CSS3 permite incluir varias imágenes de fondo, además de elegir las características de cada una ellas.

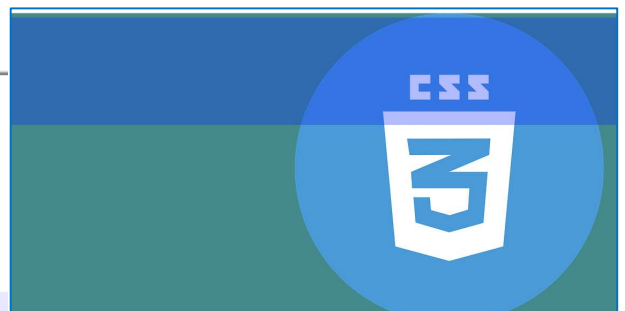
```
body{
  background-image:url("css.png"), url("html5.png");
  background-repeat: no-repeat, no-repeat;
  background-position:right, left;
}
```



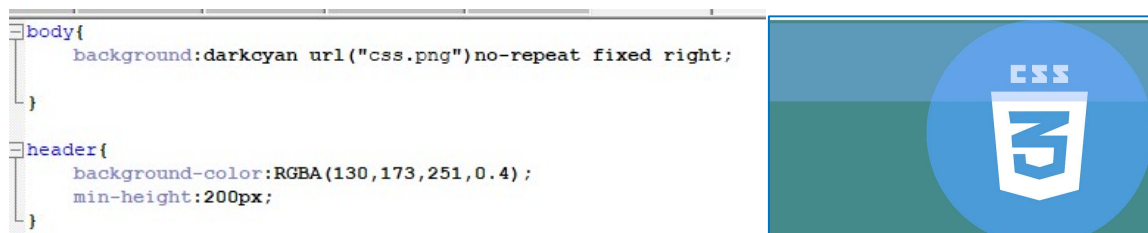
En CSS 3 podemos indicar la opacidad del fondo con la propiedad **opacity** y los valores del 0 a 1. Podemos hacer que el encabezado aparezca con un 40% de opacidad por encima del fondo del cuerpo de la página.

```
body{
  background:darkcyan url("css.png")no-repeat fixed right;
}

header{
  background-color:blue;
  opacity:0.3;
  min-height:200px;
}
```



Si optamos por poner el fondo en rgba, incluiremos el porcentaje de rojo, verde, azul y la opacidad del color elegido en una sola línea.



TEXTO

Para dar estilo al texto tenemos varias propiedades como:

- **font-family** que selecciona la Fuente
- **font-size**: para el tamaño, se suele indicar en pixel.
- **font-weight**: para el grosor de la letra, para negrita usaremos el valor **bold**.
- **Font-style**, los valores que podemos usar son **italic**, para letra cursiva, **oblique**. En ambos casos la letra está inclinada, italic es una cursiva real, mientras oblique simula la cursiva cuando la fuente no tiene modo itálico.
- **Font-variant**, para convertir a letras mayúsculas de tamaño pequeño con el valor **small-caps**.
- **Font-display**: indica al navegador que debe hacer antes de mostrar una tipografía hasta que se carga (existe la posibilidad de personalizar la fuente descargándola e indicando la url), los valores son **auto**, (texto invisible), **block** (se muestra sin estilo), **swap** (muestra la siguiente tipografía cargada por un momento), **fallback** (muestra la siguiente tipografía cargada, y deja esa si no se carga la elegida), **optional** (muestra la siguiente tipografía si la elegida tarda mucho o hay algún problema).
- **text-align**, para la alinear el texto, los valores con **right**, **center**, **left** y **justify**.
- **Text-decoration**: subraya (**underline**) o tacha (**line-through**) el texto
 - También podemos elegir el color de la línea, **text-decoration-color**
 - Estilo de la línea, siendo sólida (**solid**), ondulada (**wavy**) o discontinua (**dashed**), **text-decoration-style**
 - El grosor de la línea, **text-decoration-thickness**
- **Text-transform**: para transformar un texto a mayúsculas (**uppercase**), minúsculas (**lowercase**), o con la primera letra mayúscula y el resto minúsculas (**capitalize**)
- **Text-shadow**: añade sombra al texto, donde indicaremos la posición horizontal, vertical, y como opcionales están el radio de desenfoque y el color de la sombra (las posiciones y el radio se indicarán en pixeles, y el color el hexadecimal).
- **Text-indent**: para sangrar la primera línea del párrafo, el sangrado se manca en píxeles.
- **color**, seguido del nombre, o código del color del texto.
- **Line-height**, para el interlineado, los más comunes son **1**, **1.15** y **1.5**.
- **Vertical-align**: permite alinear el texto en su vertical con los valores **top**, **middle** y **bottom**, pero no siempre funciona bien., también tiene los valores **baseline**, **text-top**, **text-bottom**, **sub**, **super**, para alinear respecto a líneas, texto y scripts. El resultado varía mucho según el navegador.

```
font-family:"Bahnschrift SemiBold";
font-size:22px;
font-style:italic;
font-weight:bold;
font-variant: small-caps;
color:teal;
text-align:center;
text-decoration:teal underline wavy;
text-shadow:2px 2px 4px #191970;
line-height:1,5;
```



CAJAS

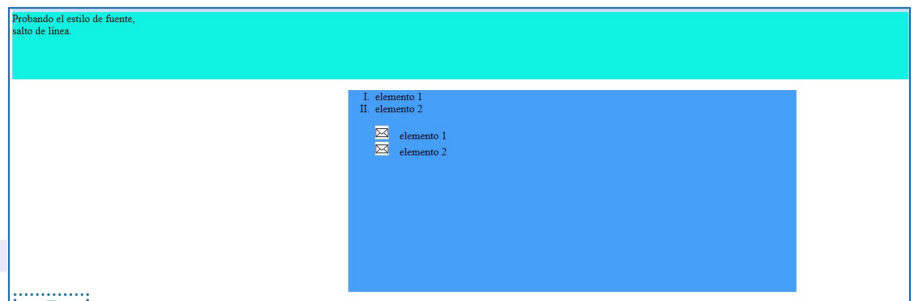
Las cajas nos permiten distribuir los elementos por la página. Para cada división podremos usar las propiedades:

- **Tamaño:** **width** (ancho) y **height** (alto), indicando pixeles o porcentaje sobre el contenedor.
- **Fondo,** con la propiedad **background**.
- **Borde:** **border**, indicando grosor, el estilo del borde con los valores más comunes como **dotted** (puntos), **dashed** (guiones), **solid** (línea normal), **double** (doble línea), y el color.
- **Relleno y márgenes:** **padding** (margen interno, relleno entre el contenido y el borde), **margin** (margen externo, distancia entre la imagen y el resto de componentes. Por defecto los elementos tienen márgenes distintos de cero, si queremos que todos los bloques de la página estén juntos, sin espacios entre ellos, podemos usar el selector general * e indicarle en el estilo que los márgenes externos e internos sean cero.

```
*{  
    margin:0px;  
    padding:0px;  
}
```

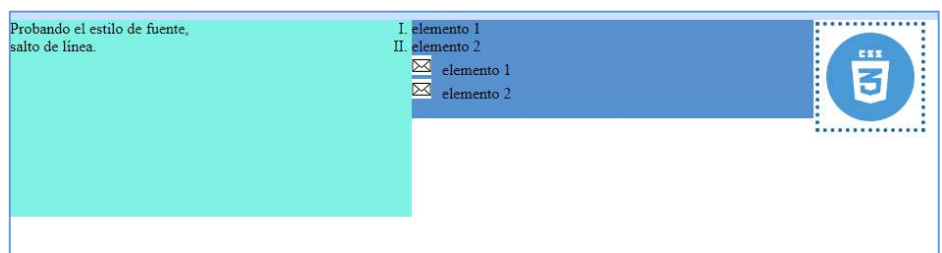
- **Posición:** **position**, donde tendremos que indicar las características de tipo con los valores, **static** (estático no se desplaza), **relative** (relative, toma como referencia la esquina superior izquierda de su contenedor), **absolute** (absolute, toma como referencia la esquina superior izquierda de la ventana del navegador) y **fixed** (fijo, toma como referencia la esquina superior derecha de la ventana del navegador).
- **Desplazamiento:** con las propiedades **left** (izquierda), **right** (derecha), **top** (superior) y **bottom** (inferior), podemos indicar el desplazamiento de la caja en pixeles o porcentajes.

```
#caja1{  
width:100%;  
height:100px;  
background-color:#10F3E2 ;  
position:relative;  
top:0px;  
}  
#caja2{  
width:50%;  
height:300px;  
background-color:#46A0FA ;  
position:relative;  
left:500px;  
top:0px;  
}
```



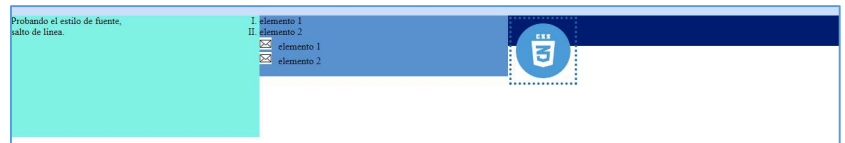
- **Flote:** **float** esta propiedad permite que las cajas floten colocándose según el valor dado (**left**, **right**, **top**, **bottom**). Si varias cajas tienen float left, la primera se colocará a la izquierda del todo, y la siguiente todo a la izquierda posible en el espacio libre dejado por la primera.

```
#caja1{  
width:30%;  
height:200px;  
background-color:#10F3E2 ;  
float:left;  
}  
#caja2{  
width:30%;  
height:100px;  
background-color:#5290CF ;  
float:left;  
}
```



- Fijar: **clear**, si queremos poner un componente debajo de las cajas que están flotantes debemos usar la propiedad clear con el valor **both**, lograremos que la siguiente caja no aparezca con las flotantes superpuestas, si no debajo de ellas.

```
#caja1{
width:30%;
height:200px;
background-color:#10F3E2 ;
float:left;
}
#caja2{
width:30%;
height:100px;
background-color:#5290CF ;
float:left;
}
footer{
background-color:#191970;
min-height:50px;
}
```



Con **clear:both**;

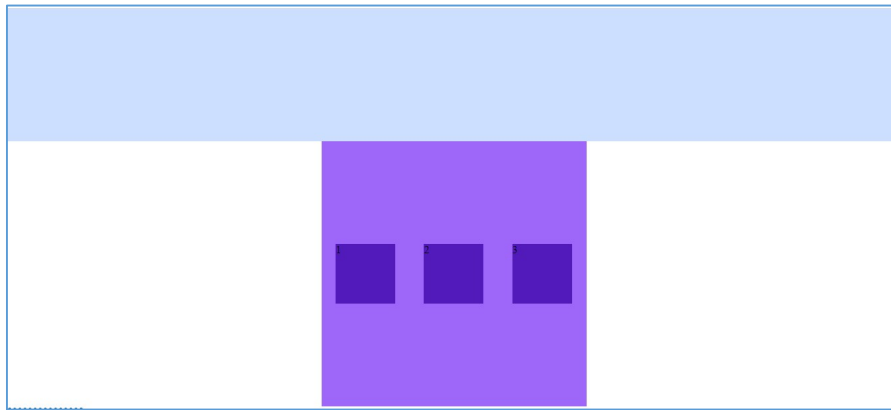
```
footer{
clear:both;
background-color:#191970;
min-height:50px;
}
```



- Cajas flexibles: con **display:flex**, indico que las cajas son flexibles, se adaptan al tamaño disponible. Normalmente usaremos una caja contenedor con cajas en su interior. Podremos orientar las cajas horizontalmente con **flex-direction:row**, o verticalmente con **column**. La propiedad **flex-wrap:wrap**, permite que las cajas del interior cambien de línea si no caben en la misma. Los márgenes automáticos de las cajas consiguen que se centren.

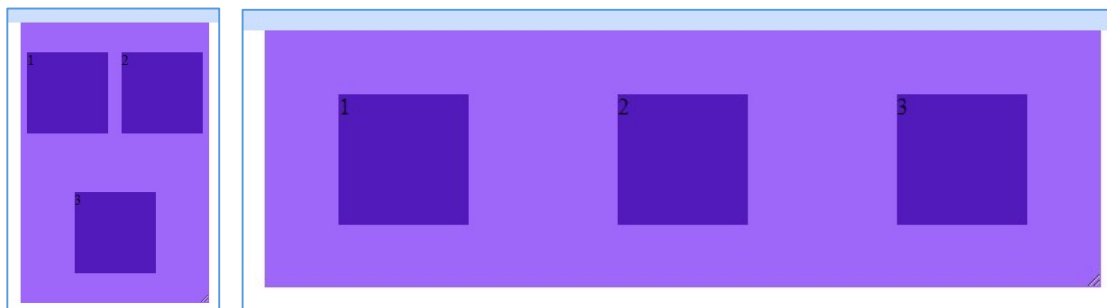
```
<div id="caja1">
<div class="cajita">1</div>
<div class="cajita">2</div>
<div class="cajita">3</div>
</div>
```

```
#caja1{
width: 400px;
height:400px;
background-color:#BE61FA ;
display:flex;
margin:auto;
flex-direction:row;
flex-wrap:wrap;
}
.cajita{
width: 90px;
height:90px;
background-color:#7304BB ;
margin:auto;
}
```



- **Redimensionar:** `resize`, permite que le usuario cambien el tamaño de la caja horizontalmente (**horizontal**), verticalmente (**vertical**) o en ambos sentidos (**both**). Debe ir acompañada de la propiedad `overflow: auto`, para que el contenido se adapta el tamaño del contenedor. El contenedor indica al usuario que puede cambiar el tamaño con líneas diagonales en la esquina inferior derecha.

```
#cajal{
  width: 400px;
  height:400px;
  background-color:#BE61FA ;
  display:flex;
  margin:auto;
  flex-direction:row;
  flex-wrap:wrap;
  resize:both;
  overflow:auto;
}
```



LISTAS

El formato propio de las listas se configura con las propiedades:

list-style-type, para el tipo de viñeta, los valores que puede tomar son para listas no ordenadas: **disc** (punto), **circle** (circulo) y **square** (cuadrado), para las listas ordenadas: **decimal** (números), **decimal-leading-zero** (números con cero delante), **lower-roman** (números romanos en minúsculas), **upper-roman** (números romanos en mayúsculas), **lower-latin** (letras minúsculas), **upper-latin** (letras mayúsculas), **lower-greek** (letras griegas), **none** (sin viñeta).

List-style-position, para que todas las líneas de un mismo elemento empiecen en la misma posición (**inside**), o que las siguientes líneas empizen más a la derecha(**outside**).



```
ul {
  list-style-type:square;
  list-style-position: inside;
}
ol {
  list-style-type:upper-roman;
  list-style-position: outside;
}
```

I. elemento 1
II. elemento 2

- elemento 1
- elemento 2

CSS permite que usemos cualquier imagen como viñeta de una lista con la propiedad `list-style-image`, indicando la url de la imagen.

```
ul {
  list-style-image:url("sobre.png");
  list-style-position: inside;
}
```

 elemento 1
 elemento 2

Si queremos indicar varias propiedades podemos hacerlos usando la propiedad genérica `list-style`, seguidos del tipo, posición e imagen.

```
list-style:url("sobre.png") inside;
```

TABLAS

Para dar formato con CSS a las tablas contamos con las propiedades de ancho (`width`) y alto(`height`) donde podremos indicar el tamaño como pixeles o porcentajes, todas las vistas para el texto, y otras como:

Border-collapse con los valores `collapse` para que los bordes de las celdas de la tabla aparezcan juntos, o `separate` para que se vean separados.

Border-spacing, donde indicaremos el espacio entre bordes.

Podremos usar las propiedades `padding` (margen interno, relleno entre el contenido y el borde), `margin` (margen externo, distancia entre la imagen y el resto de componentes) y borde (`border`), para la que debemos indicaremos el grosor, el estilo del borde con los valores más comunes como `dotted` (puntos), `dashed` (guiones), `solid` (línea normal), `double` (doble línea), y el color.

Si queremos dar formato a las celdas, usaremos el selector descendiente, es decir después del nombre del estilo de la tabla, indicaremos la celda.

```
.tabla1{
  width:300px;
  height:100px;
  text-align: center;
  border: 2px dashed purple;
  border-spacing:0px; /*espacio entre bordes*/
  border-collapse:collapse; /* superposición de los bordes*/
  padding:0.1em;
  background:#FFA07A;
}
.tabla1 td{
  border:1px solid blue;
}
```

celda 1	celda 2
celda 3	celda 4

La propiedad `empty-cell` permite ocultar (hide) o mostrar(show) las celdas vacías. No puede combinarse con la opción de `collapse`. También podemos modificar la posición del título de la tabla con `caption-side` con los valores `top` o `bottom`.

```
<table class="tabla1">
  <caption>Título tabla</caption>
  <tr>
    <td>celda 1</td>
    <td>celda 2</td>
  </tr>
  <tr>
    <td>celda 3</td>
    <td></td>
  </tr>
</table>
```

```
.tabla1{
  width:300px;
  height:100px;
  text-align: center;
  border-spacing:0px;
  padding:0.1em;
  empty-cells:hide;
  caption-side:bottom;
}
.tabla1 td{
  border:1px solid blue;
}
```

celda 1	celda 2
celda 3	
Título tabla	

IMÁGENES

Con CSS se puede dar formato a todas las imágenes usando la etiqueta `img` en el estilo, o a una concreta usando su identificador.

```
</img>
```

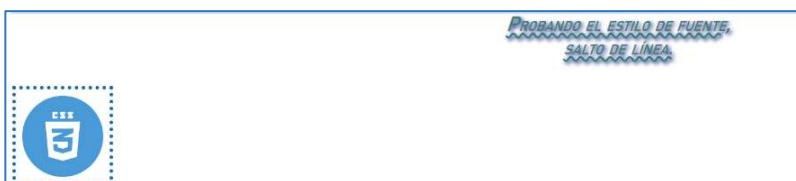
Las propiedades más comunes para las imágenes son la anchura (**width**) y la altura (**height**), que se concretan usando pixeles, porcentajes.

Podremos usar las propiedades **padding** (margen interno, relleno entre el contenido y el borde), **margin** (margen externo, distancia entre la imagen y el resto de componentes) y borde (**border**), para la que debemos indicaremos el grosor, el estilo del borde con los valores más comunes como **dotted** (puntos), **dashed** (guiones), **solid** (línea normal), **double** (doble línea), y el color.

A las imágenes, como a las tablas podemos aplicarles las propiedades, **margin, padding y border**.

Además del modo acortado, también se puede definir cada uno de los márgenes o bordes de la imagen (derecho, izquierdo, inferior y superior) con la propiedad específica, **padding-top, margin-bottom, border-right-width, border-top-color, border-left-style**.

```
#css {  
  width:100px;  
  height:100px;  
  border: 4px dotted #0469A5 ;  
  padding: 2px;  
  margin: 2px;  
  float:left;  
}
```



La propiedad **float** permite posicionar la imagen respecto al texto, con los valores **left** y **right**.

ENLACES

Los enlaces suelen ser texto o imágenes, por ello las propiedades de estilo usadas son las explicadas para estos elementos. (una de las que se suele usar es **text-decoration: none**; para quitar el subrayado típico del enlace)

Debemos recordar que existen pseudoclases exclusivas de los enlaces, por lo que podemos dar estilo diferentes o iguales a los enlaces según se hayan o no visitado. (**:link**, **:visited**). Además puede interesarnos dar distintos formatos cuando se pase el ratón por encima o se active, (**:hover**, **:active**)

CURSOR

Podemos modificar el cursor del ratón con la propiedad **cursor**, podemos modificarlo usando un icono propio con su url, o usar las formas preestablecida **auto** (puntero de texto), **cell** (cruz rellena), **col-resize** (flechas a izquierda y derecha con líneas paralelas verticales en medio) **crosshair** (cruz), **copy** (flecha con un +), **default** (flecha), **grab** (mano abierta), **grabbing** (mano agarrando) **pointer** (mano señalando), **move** (flechas en cruz), **no-drop** y **not-allowed** (prohibido), **none** (sin cursor), **ew-resize** (flecha puntas en extremos horizontal), **ns-resize** (flecha puntas en extremos vertical), **nesw-resize** (flecha puntas en extremos de noreste a suroeste), **nwse-resize** (flecha puntas en extremos de noroeste a sureste), **row-resize** (flechas hacia arriba y abajo con líneas paralelas horizontales en medio) **text** (puntero de texto), **wait** (circulo girando), **help** (flecha con interrogación), **progress** (flecha con circulo de espera), **zoom-in** (lupa con +), **zoom-out** (lupa con menos).

Esta propiedad se combina con las acciones del ratón, como son las pseudoclases **hover**, **focus**, o **active**.

ANIMACIONES

Con CSS se pueden hacer animaciones. En el documento html deberemos identificar los elementos que queremos animar y en el documento CSS configuraremos las animaciones. Para hacer una animación debemos usar la regla de los fotogramas clave **@keyframe**.

Al inicio del CSS crearemos la regla indicando los fotogramas con porcentaje (si solo hay 2 podemos usar from para el inicial y to para el final), para cada uno de ellos debemos indicar su posición (**left**, **right**, **top** o **bottom** en pixeles) y podemos añadirle estilo propio como color de fondo.

Después crearemos el estilo del elemento identificado, además del estilo general como tamaño, fondo, borde..., tendremos que incluir propiedades propias de las animaciones.

El elemento debe estar posicionado (**position: relative** o **absolute**) para poder moverlo por la página.

- **Animation-name**: nombre del keyframe. Es obligatorio.
- **Animation-duration**: duración de la animación (en segundos "s" o milisegundos "ms"). Es obligatorio.
- **animation-timing-function**: controla la velocidad de la animación **linear** (velocidad constant todo el tiempo), **ease** (inicia lento, se acelera y termina lenta), **ease-in** (inicia lento y termina rápido), **ease-out** (inicia rápido y termina lento).
- **Animation-delay**: tiempo que tarde en empezar la animación.
- **Animation-iteration-count**: número de veces que se repetirá, desde **0** hasta infinito (**infinite**).
• **Animation-direction**: tiene los valores: **normal** (solo se mueve en un sentido) o **alternate** (va y vuelve).

Para estos valores podemos usar la forma corta, **animation**, seguido de las opciones elegidas en el orden en que se han nombrado.

```
@keyframes avisoA {
  0% { left: 0px; background-color:blue}
  100% { left: 600px;background-color:green }
}

#aviso {
  width: 150px;
  height: 50px;
  position: relative;
  animation: avisoA 10s ease-out 3 alternate;
  /*nombre del keyframe, duración, velocidad, repeticiones y recorrido*/
}

#aviso > p{
  font: 20px "algerian";
  color: white;
  text-align:center;
}
```

**AVISO
IMPORTANTE**

**AVISO
IMPORTANTE**

- **animation-play-state**: indica si la animación está active (**running**), o pausada (**paused**). Suele usarse combinado con botones para la interacción del usuario.

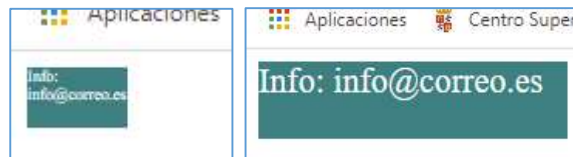
Otro tipo de animaciones son las **transiciones**, simulamos la animación de un elemento cambiando propiedades como tamaño o color en función del estado en el que se encuentre el elemento, solemos usar la pseudoclase **:hover**;

Las propiedades para hacer transiciones son similares a las de las animaciones:

- **Transition-property** indica a que propiedades se le va a aplicar la transición, con los valores **none**, **all** o con la o las propiedades concretas.
- **Transition-duration**: tiempo en segundos (s) o milisegundos (ms)
- **Transition-timing-function**: con los valores **linear**, **ease**, **ease-in** y **ease-out**.
- **Transition-delay**: retraso de la animación.

```
<div id="info"> Info: info@correo.es </div>
```

```
#info{
  width:50px;
  height:30px;
  background-color:teal;
  color:white;
  font-size:8px;
  transition-property: all;
  transition-duration: 2s;
  -webkit-transition-property: all; -webkit-transition-duration: 2s
}
#info:hover{
  width: 200px;
  height:50px;
  font-size:22px;
}
```



Otra forma de crear animaciones es usar las **transformaciones**, que cambiarán el aspecto de un elemento.

Las propiedad **transform**, puede usar varios métodos:

- **Translate**, traslada el objeto a la posición indicada por los eje x y del eje y. Este método tiene las opciones de **translateX** (cambia la posición solo en el eje x) y **translateY** (cambia la posición solo en el eje y). También se puede trasladar en las 3 dimensiones, **translate3d**, o en el eje z **translateZ**.
- **Rotate**: rota la imagen los grados indicados seguido de deg. Admite rotaciones en el eje X (**rotateX**), y en eje Y (**rotateY**)
- **Scale**: escala la imagen indicando la proporción que va a aumentar la anchura y la altura mediante números.
- **Skew**, inclina la imagen en los ejes x e y, o solo en el x (**skewX**) o solo en el y (**skewY**).

```
#fotol{
  width:30px;
  height:30px;
}
#fotol:hover {
  transform:translate(700px,200px) rotate(45deg) scale(10,10) skewY(40deg);
}
```



Si combinamos las transiciones con las transformaciones y con las animaciones, podremos crear animaciones más complejas.

```
#fotol{
  width:30px;
  height:30px;
  transition:all 2s;
}
#fotol:hover {
  transform:translate(700px,200px) rotate(360deg) scale(10,10) skewY(40deg);
}
```

En este ejemplo veríamos como pasamos de la foto pequeñita a la izquierda, a la grande y deformada que va girando 360° en el desplazamiento.

```
@keyframes css {
  from { transform:rotate(0deg) scale(1,1); }
  to { transform:rotate(360deg) scale(5,5); }
}
#fotol {
  width: 50px;
  height: 50px;
  position: relative;
  left:50%;
  top:100px;
  animation: css 5s linear infinite normal;
}
```



En este caso, veremos como la imagen gira 360° mientras aumenta 5 veces su tamaño.

EFFECTOS

Con la propiedad **filter** podemos aplicar filtros. Esta opción combina muy bien con las animaciones y las pseudoclases.

Tenemos los siguientes filtros:



(imagen original)

- **Grayscale** podemos modificar el color desde el original (0%) hasta blanco y negro (100%)

`filter: grayscale(70%);`



- **Sepia**, podemos modificar el color desde el original (0%) hasta el sepia(100%)

`filter: sepia(75%);`



- **Blur**, difumina la imagen, el valor será el radio de desenfocado.

`filter: blur(3px);`



- **Saturate**, satura la imagen con blanco y negro (0%), original (100%) o sobresaturado (>100%)

`filter: saturate(60%);`



- **Opacity**, transparencia desde transparencia (0%) hasta opaco(100%)

`filter: opacity(19%);`



- **Brightness**, modifica el brillo desde imagen negra(0%) pasando por original (100%) hasta quemada (>100%)

`filter: brightness(0.5);`



- **contrast** cambie el contraste desde gris (0%), original (100%) a sobresaturado (>100%)

`filter: contrast(170%);`



- **hue-rotate**, altera el color rotando en el círculo cromático, desde rojo (1deg), amarillo (90deg), azul (270deg) a original (0 o 360).

`filter: hue-rotate(150deg);`



- **Invert**, invertir el color de la imagen desde original (0%) hasta 100%(invertido).

`filter: invert(40%);`



- **drop-shadow**, añade una sombra con forma idéntica a la imagen, donde indicaremos la posición horizontal, vertical, el tamaño y el color.

`filter: drop-shadow(5px 5px 10px #F08080);`








CÓDIGO DE COLORES:


















	INDIANRED	#CD5C5C
	LIGHTCORAL	#F08080
	SALMON	#FA8072
	PINK	#FFC0CB
	DARKSALMON	#E9967A
	LIGHTPINK	#FFB6C1
	LIGHTSALMON	#FFA07A
	HOTPINK	#FF69B4
	CRIMSON	#DC143C
	DEEPPINK	#FF1493
	RED	#FF0000
	MEDIUMVIOLETRED	#C71585
	FIREBRICK	#B22222
	PALEVIOLETRED	#DB7093
	DARKRED	#8B0000
	LIGHTSALMON	#FFA07A
	CORAL	#FF7F50
	TOMATO	#FF6347
	ORANGERED	#FF4500
	DARKORANGE	#FF8C00
	ORANGE	#FFA500

	GOLD	#FFD700			
	YELLOW	#FFFF00		LAVENDER	#E6E6FA
	LIGHTYELLOW	#FFFFE0		THISTLE	#D8BFD8
	LEMONCHIFFON	#FFFACD		PLUM	#DDA0DD
	LIGHTGOLDENRODYELLOW	#FAFAD2		VIOLET	#EE82EE
	PAPAYAWHIP	#FFEFD5		ORCHID	#DA70D6
	MOCCASIN	#FFE4B5		FUCHSIA	#FF00FF
	PEACHPUFF	#FFDAB9		MAGENTA	#FF00FF
	PALEGOLDENROD	#EEE8AA		MEDIUMORCHID	#BA55D3
	KHAKI	#F0E68C		MEDIUMPURPLE	#9370DB
	DARKKHAKI	#BDB76B		REBECCAPURPLE	#663399
	BLUEVIOLET	#8A2BE2			
	DARKVIOLET	#9400D3			
	DARKORCHID	#9932CC			
	DARKMAGENTA	#8B008B			
	PURPLE	#800080			
	INDIGO	#4B0082			
	SLATEBLUE	#6A5ACD			
	DARKSLATEBLUE	#483D8B			
	MEDIUMSLATEBLUE	#7B68EE			

			 FORESTGREEN	#228B22
 GREENYELLOW	#ADFF2F	 GREEN		#008000
 CHARTREUSE	#7FFF00	 DARKGREEN		#006400
 LAWNGREEN	#7CFC00	 YELLOWGREEN		#9ACD32
 LIME	#00FF00	 OLIVEDRAB		#6B8E23
 LIMEGREEN	#32CD32	 OLIVE		#808000
 PALEGREEN	#98FB98	 DARKOLIVEGREEN		#556B2F
 LIGHTGREEN	#90EE90	 MEDIUMAQUAMARINE		#66CDAA
 MEDIUMSPRINGGREEN	#00FA9A	 DARKSEAGREEN		#8FBC8B
 SPRINGGREEN	#00FF7F	 LIGHTSEAGREEN		#20B2AA
 MEDIUMSEAGREEN	#3CB371	 DARKCYAN		#008B8B
 SEAGREEN	#2E8B57	 TEAL		#008080

	AQUA	#00FFFF		POWDERBLUE	#B0E0E6
	CYAN	#00FFFF		LIGHTBLUE	#ADD8E6
	LIGHTCYAN	#E0FFFF		SKYBLUE	#87CEEB
	PALETURQUOISE	#AFEEEE		LIGHTSKYBLUE	#87CEFA
	AQUAMARINE	#7FFFD4		DEEPSKYBLUE	#00BFFF
	TURQUOISE	#40E0D0		DODGERBLUE	#1E90FF
	MEDIUMTURQUOISE	#48D1CC		CORNFLOWERBLUE	#6495ED
	DARKTURQUOISE	#00CED1		MEDIUMSLATEBLUE	#7B68EE
	CADETBLUE	#5F9EA0		ROYALBLUE	#4169E1
	STEELBLUE	#4682B4		BLUE	#0000FF
	LIGHTSTEELBLUE	#B0C4DE		MEDIUMBLUE	#0000CD
	DARKBLUE	#00008B			
	NAVY	#000080			
	MIDNIGHTBLUE	#191970			

	CORNSILK	#FFF8DC		
	BLANCHEDALMOND	#FFEBCD		
	BISQUE	#FFE4C4		
	NAVAJOWHITE	#FFDEAD		
	WHEAT	#F5DEB3		
	BURLYWOOD	#DEB887		PERU #CD853F
	TAN	#D2B48C		CHOCOLATE #D2691E
	ROSYBROWN	#BC8F8F		SADDLEBROWN #8B4513
	SANDYBROWN	#F4A460		SIENNA #A0522D
	GOLDENROD	#DAA520		BROWN #A52A2A
	DARKGOLDENROD	#B8860B		MAROON #800000

	WHITE	#FFFFFF		
	SNOW	#FFFAFA		
	HONEYDEW	#F0FFF0		
	MINTCREAM	#F5FFFA		
	AZURE	#F0FFFF		
	ALICEBLUE	#F0F8FF		FLORALWHITE #FFFAF0
	GHOSTWHITE	#F8F8FF		IVORY #FFFFF0
	WHITESMOKE	#F5F5F5		ANTIQUEWHITE #FAEBD7
	SEASHELL	#FFF5EE		LINEN #FAF0E6
	BEIGE	#F5F5DC		LAVENDERBLUSH #FFF0F5
	OLDLACE	#FDF5E6		MISTYROSE #FFE4E1

	GAINSBORO	#DCDCDC
	LIGHTGRAY	#D3D3D3
	SILVER	#C0C0C0
	DARKGRAY	#A9A9A9
	GRAY	#808080
	DIMGRAY	#696969
	LIGHTSLATEGRAY	#778899
	SLATEGRAY	#708090
	DARKSLATEGRAY	#2F4F4F
	BLACK	#000000

FUENTES CONSULTADAS.

Páginas consultadas para realizar los apuntes:

- <https://brandominus.com/>
- <https://desarrolloweb.com/>
- <https://developer.mozilla.org/>
- <https://htmlcolorcodes.com/>
- <https://lenguajecss.com>
- <https://lenguajehtml.com/>
- <https://processing.org/>
- <https://uniwebsidad.com/>
- <https://www.arsys.es/blog>
- <http://www.htmlquick.com>
- <https://www.w3schools.com/>