

# HTML



# CSS



# HTML 5

# CSS 3

Omar del Río García

Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

 Generalitat  
de Catalunya





## CONTENIDO

Introducción.....	6
HTML 5 .....	8
Etiquetas.....	8
Atributos.....	8
Estructura Global .....	9
Tipo de documento .....	9
body .....	12
Organización del documento.....	12
header .....	13
nav.....	13
footer .....	13
aside .....	13
main .....	13
section.....	13
Definición de contenidos .....	14
article .....	14
details.....	14
summary .....	14
figure .....	14
figcaption .....	14
details.....	14
summary .....	14
address .....	14
div.....	14
Tipos de contenidos principales .....	17
Parágrafo.....	17
Encabezamiento.....	17
Listas.....	17
Citas.....	18
Separadores .....	19
Tipos de contenidos.....	19
Resaltado de texto .....	19
Citas de texto .....	19



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

Línea base .....	20
Control de canviós .....	20
Salto de línia .....	20
Enllaços .....	20
Barras de progrés .....	21
span .....	22
Contenidos multimédia .....	22
Imágenes .....	22
Mapas de imágenes .....	23
Audio .....	24
Vídeo .....	25
Ventanas de diálogo o emergentes .....	25
Tablas .....	26
Marcos .....	28
Formularios .....	29
Organización de los campos .....	29
Tipos de campos .....	30
Envío del formulario .....	32
Atributos globales .....	33
CSS 3 .....	35
Introducir el CSS .....	35
Selectores .....	35
Compatibilidad con versiones anteriores de los navegadores .....	39
Compatibilidad con las nuevas propiedades .....	39
Propiedades .....	39
Modelo de caja .....	39
Contenido generado, numeración automática y listas .....	43
Colores y Fondo .....	43
Fuentes .....	45
Texto .....	46
Tablas .....	46
Interfaz de usuario .....	46
Media queries .....	46
Transform .....	50



Transiciones .....	51
Animaciones .....	52
Maquetación.....	53
Float .....	53
Flexbox.....	55
Grid.....	59
Columnas.....	62
Variables .....	63
Funcionamiento de las variables en CSS3 .....	63
Modo Claro-Oscuro .....	63



**Generalitat  
de Catalunya**

Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)



SERVICIO PÚBLICO  
DE EMPLEO ESTATAL  
**SEPE**  
SERVEI PÚBLIC  
D'OCCUPACIÓ ESTATAL

## INTRODUCCIÓN

En el siguiente manual encontrarás una breve guía de cómo utilizar los códigos webs necesarios para poder realizar una página o aplicación web: los lenguajes HTML y CSS, en sus últimas versiones. Estos dos lenguajes han evolucionado mucho en los últimos años, pero su uso y filosofía se mantiene igual, por tanto, estudiar HTML 5 y CSS 3 es estudiar sus versiones anteriores más las novedades.

Una de las principales características de estos lenguajes es que son código libre, por tanto, no es necesario utilizar ningún software especial para utilizarlos: sólo hace falta un editor de texto sin formato y un navegador.

El editor de texto podemos utilizar el "*bloc de notas*" de Microsoft Windows, pero es recomendable utilizar uno que resalte el código en colores para identificar las partes del mismo e identificar rápidamente errores de sintaxis.

A estos programas, que normalmente añaden diferentes funcionalidades además de la de editor de código, se les denominan **entorno integrado de desarrollo** o **IDE** (en inglés *integrated development environment*). A continuación listo unas recomendaciones:

- **Visual Studio Code:** editor gratuito disponible para todas las plataformas. Es muy potente e interpreta muchos lenguajes de programación; además cuenta con una consola para ejecutar órdenes de otros sistemas.  
**URL:** <https://code.visualstudio.com/>
- Es muy fácil de instalar y dispone de muchos complementos para ampliar su funcionamiento y traducción a muchos idiomas.  
**URL:** <http://notepad-plus-plus.org/>
- **Brackets:** otro editor gratuito de Adobe para todas las plataformas. Es muy sencillo y básico, pero con una cuenta de Adobe tienes acceso a más funcionalidades.  
**URL:** <http://brackets.io/>
- **Sublime Text:** es un editor para todas las plataformas; te puedes descargar la versión gratuita pero tiene publicidad para que lo compres.  
**URL:** <http://www.sublimetext.com/>
- **Atom:** es un editor para todas las plataformas .  
**URL:** <https://atom.io/>
- **NetBeans:** es un editor para todas las plataformas .  
**URL:** <https://netbeans.apache.org/>
- **Dreamweaver:** editor de Adobe muy completo por todas las plataformas pero de pago.  
**URL:** <http://www.adobe.com/es/products/dreamweaver.html>

Respecto al navegador, hay que tener presente que es necesario tener siempre actualizado, porque HTML 5 y CSS 3 todavía no son lenguajes estándar, y no lo serán hasta que estos lenguajes estén bien implantados.



**Generalitat  
de Catalunya**



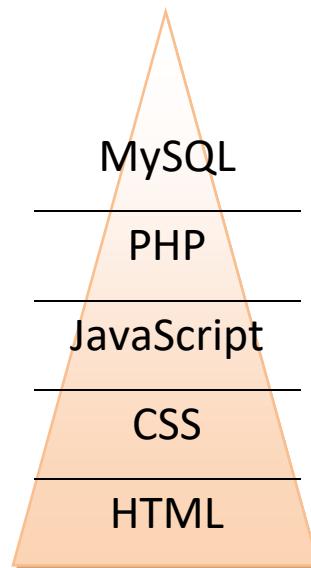
Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

De la estandarización de estos lenguajes y otros muchos relacionados con el mundo web, se encarga el **W3C**: una comunidad internacional que se encarga de desarrollar los estándares web.

El objetivo de este manual no es mostrar todas las opciones posibles porque son lenguajes vivos que cambian continuamente, por tanto, recomiendo consultar bloques, foros y otras web de referencia como por ejemplo <http://www.w3schools.com/> o <https://developer.mozilla.org/>.

De forma rápida, podemos listar los lenguajes web como una pirámide, donde cada uno de ellos complementa el anterior:

- En la base tenemos el **HTML**, que nos permite estructurar los contenidos por su representación.
- El **CSS** complementa al HTML para darle estilo y diseño, una mejor apariencia en la estructura anterior.
- **JavaScript** da dinamismo y efectos especiales al HTML y al CSS. La suma de los tres lenguajes se denomina DHTML, y su principal característica es que sólo es necesario un navegador para ver los resultados.
- El **PHP** es un lenguaje de servidor, por tanto, permite aumentar la seguridad y utilizar recursos de forma independiente en el navegador del usuario.
- **MySQL** es una tecnología de base de datos, que le permite al PHP almacenar datos y contenidos para recuperarlos posteriormente; es la base para desarrollar herramientas web como gestores de contenidos.



## HTML 5

El lenguaje **HTML** (acrónimo de *Hyper Text Markup Language*, en castellano, "lenguaje de marcado de hipertexto") es un lenguaje de marcado que deriva del SGML diseñado para estructurar textos y relacionarlos en forma de hipertexto. Gracias a Internet y los navegadores web, se ha convertido en uno de los formatos más populares que existen para la construcción de documentos para la web.

Es un lenguaje porque nos permite decirle a un navegador qué debe mostrar; es de marcado porque para ello ponemos unas marcas determinadas (las estandarizadas por el W3C) dentro del código; y es de hipertexto porque está pensado para mostrarse en un navegador y realizar enlaces entre documentos, aunque hoy en día se utiliza en muchas más aplicaciones que no son web.

Estas marcas son específicas, como acabo de comentar, y en los siguientes capítulos voy a explicar las principales marcas y su uso. Hay otros lenguajes donde las marcas se pueden inventar, como el **XML** (del inglés *eXtensible Markup Language*, en castellano "lenguaje de marcas extensible"), pero éstos no los trataremos aquí.

### ETIQUETAS

Las etiquetas son el pilar fundamental del código: son las "marcas" a las que se refiere el nombre del lenguaje. Sirven para definir una estructura y, a su vez, da significado a su contenido, diciéndole a los navegadores y buscadores qué tipo de contenido hay dentro.

La estructura básica es la siguiente: un nombre de etiqueta entre símbolos "<" y ">".

```
<etiqueta>contenido</etiqueta>
```

El nombre "etiqueta" hace referencia a un nombre específico que da una estructura y significado al "contenido" que aparece a continuación. Las etiquetas deben cerrarse para finalizar el efecto de éstas, y se cierra poniendo una barra "/" delante del nombre de la etiqueta.

Algunas etiquetas no es necesario cerrarlas porque es un efecto puntual o sirve para insertar un objeto, no para estructurar un contenido.

En función de la versión del código (y la versión del navegador) podemos utilizar unas u otras, por tanto, es importante especificar el *doctype* al principio del documento.

### ATRIBUTOS

A veces una etiqueta es demasiado genérica y es necesario definir más su uso: en estos casos utilizamos atributos también específicos para cada una de ellas. Estos atributos se ponen en la etiqueta de inicio y su efecto dura hasta que ponemos la etiqueta final.

La estructura básica es la siguiente: un nombre de atributo seguido del símbolo "=" y el valor entre comillas, separado del nombre de la etiqueta por un espacio en blanco.

```
<etiqueta atributo1="valor" atributo2="valor" atributo3="valor"> contenido
</etiqueta>
```



**Generalitat  
de Catalunya**



SERVICIO PÚBLICO  
DE EMPLEO ESTATAL  
**SEPE**  
SERVICIO PÚBLICO  
D'OCUPACIÓ ESTATAL

## ESTRUCTURA GLOBAL

El código de un documento HTML tiene una estructura inicial muy definida: siempre empezamos con `<!DOCTYPE>` y continuamos con la etiqueta `<html></html>`. Esta etiqueta indica en el navegador que ahora empezamos a utilizar lenguaje HTML, y es la primera etiqueta que se abre y la última que se cierra; todo lo que hacemos a continuación va dentro.

Esta obligación de definir el lenguaje se debe a que un documento web puede tener diferentes extensiones (.html, .htm, .php, .tpl), y dentro de un documento web se puede utilizar muchos lenguajes (HTML, CSS, JS, PHP) , por tanto, hay que definir siempre el lenguaje que utilizamos.

Para definir el idioma del contenido del documento podemos añadir el atributo `lang` a cualquier etiqueta; si añadimos el atributo a la etiqueta `<html>` definimos el idioma por defecto de todo el documento. El nombre del idioma lo especificamos por el código ISO 639-1 de dos letras ([http://www.w3schools.com/tags/ref\\_language\\_codes.asp](http://www.w3schools.com/tags/ref_language_codes.asp)).

Dentro de la etiqueta `<html>` siempre van dos etiquetas de apertura y cierre: `<head>` y `<body>`

```
<!DOCTYPE html>
<html lang="es">
  <head>

  </head>
  <body>

  </body>
</html>
```

## TIPO DE DOCUMENTO

Dado que el lenguaje HTML ha sufrido evoluciones en su uso y aparición de marcas nuevas, W3C recomienda especificar en el navegador la versión que estamos utilizando. Para ello empezamos cada documento con una etiqueta DOCTYPE para indicar la versión.

### HTML 4

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

Características de la codificación:

1. Podem utilizar mayúsculas y minúsculas para las etiquetas y atributos.
2. No es necesario cerrar las etiquetas.
3. Los valores de los atributos de las etiquetas pueden ir sin comillas.
4. Los estilos se aplican con etiquetas `<font></font>`.

### XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
...
</html>
```



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

Características de la codificación:

1. Sólo se emplean las minúsculas.
2. Es necesario cerrar todas las etiquetas (<etiqueta></etiqueta>); las etiquetas que no tienen cierre se cierran con la forma <etiqueta />
3. Los valores de los atributos de las etiquetas van siempre entre comillas.
4. Las etiquetas subordinadas se tabulan.

## HTML 5

```
<!DOCTYPE html>
<html>
...
</html>
```

Características sin definir.

El estándar actual es HTML 5, aunque no tiene línea de definición porque la especificación actual se conoce como HTML Living Standard.

Si estudiamos en detenimiento la línea, podemos ver diferentes partes:

```
<!DOCTYPE html PUBLIC
```

Especifica un documento tipo html.

```
"-//W3C//DTD XHTML 1.0 Transitional//EN"
```

Según los estándares W3C, la *Definición del Tipo de Documento* es un XHTML de versión 1.0 tipo Transición, y el idioma del código es el inglés.

Los documentos pueden ser de tipo *Transitional o Strict*, según cómo utilicemos el código. El idioma del código HTML siempre está en inglés: veremos que los nombres de las etiquetas son nombres ingleses.

```
"http://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd">
```

La última parte es un url en el que se puede consultar la definición de la versión del código en la web del W3C.

## HEAD

En esta región ponemos etiquetas referentes a recursos que complementarán los contenidos del <body>, como por ejemplo los estilos CSS, funciones JavaScript, otros recursos externos y la meta información del mismo documento.

## TITLE

Una parte obligatoria dentro del <head> es la etiqueta <title>, en la que especificamos el título del documento. Al estar dentro del <head> no se verá en el contenido, sino en la ventana del navegador.

```
<!DOCTYPE html>
<html lang="ca">
  <head>
    <title>Título del documento</title>
  </head>
```



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

```
<body>  
</body>  
</html>
```

## META

La etiqueta **<meta>** es una etiqueta genérica para poder especificar cualquier tipo de información, por tanto, necesitamos atributos para definir exactamente su significado. Los atributos que utilizamos son:

- **name:** el tipo de información que queremos poner.
- **content:** el contenido de la información.
- **http-equiv:** sirve para modificar el funcionamiento del navegador por este documento.
- **charset:** para especificar el tipo de caracteres que utilizaremos en el documento.

```
<!DOCTYPE html>  
<html lang="es">  
  <head>  
    <title>Título del documento</title>  
    <meta charset="utf-8">  
    <meta name="description" content="Ejemplo de código HTML 5">  
    <meta name="keywords" content="curso, html, css">  
  </head>  
  <body>  
  </body>  
</html>
```

Al colocar el contenido dentro de los atributos, esta etiqueta no se cierra.

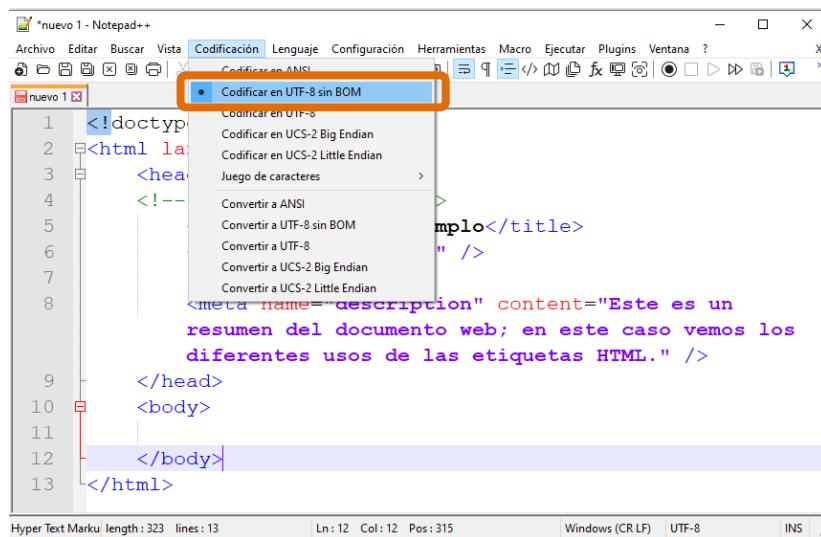
Una meta útil para adaptar nuestra web a cualquier dispositivo es lo que indica que el contenido no se adapte a la pantalla del dispositivo si no se muestre a su tamaño real (luego habrá que aplicar CSS para adaptar el contenido):

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

## CHARSET

Esta información sirve para avisar al navegador cómo están definidos los caracteres del documento, y esto lo determina el editor que hemos empleado para realizar el código. Si no se determina correctamente, los caracteres especiales ("ç", "ñ" y acentos) no se verán correctamente.

Por ejemplo, en Notepad++ la codificación del documento es en UTF-8, por tanto, lo definimos en el documento.



Algunos ejemplos de cómo codificar el documento:

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta charset="utf-8">
```

## BODY

En esta región ponemos los contenidos que queremos mostrar en la ventana del navegador. Este contenido se estructura con etiquetas, y se pueden dividir en dos grandes bloques:

- Etiquetas de bloque:** son etiquetas que rompen el flujo del texto y separan el contenido que hay previamente y del siguiente.
- Etiquetas de línea:** son etiquetas que no rompen el flujo de texto y lo dejan con el contenido anterior y posterior.

Contenido en bloque, contenido en bloque.

Contenido en bloque, contenido en bloque, contenido en línea, contenido en bloque, contenido en bloque.

Contenido en bloque, contenido en bloque.

## ORGANIZACIÓN DEL DOCUMENTO

Dentro de un documento web no todos los contenidos tienen el mismo significado o funcionalidad, por tanto, primero debemos organizar los contenidos. Independientemente del tipo de web, todas tienen una estructura similar:



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

**HEADER**

Etiqueta de bloque para especificar información introductoria del contenido principal, como títulos y logos.

**NAV**

Etiqueta de bloque para especificar dónde se encuentra el menú de navegación principal. Puede estar fuera o también dentro del <header>.

**FOOTER**

Etiqueta de bloque para especificar el final de los contenidos del documento y que normalmente contiene información sobre el propietario/autor de la web.

**ASIDE**

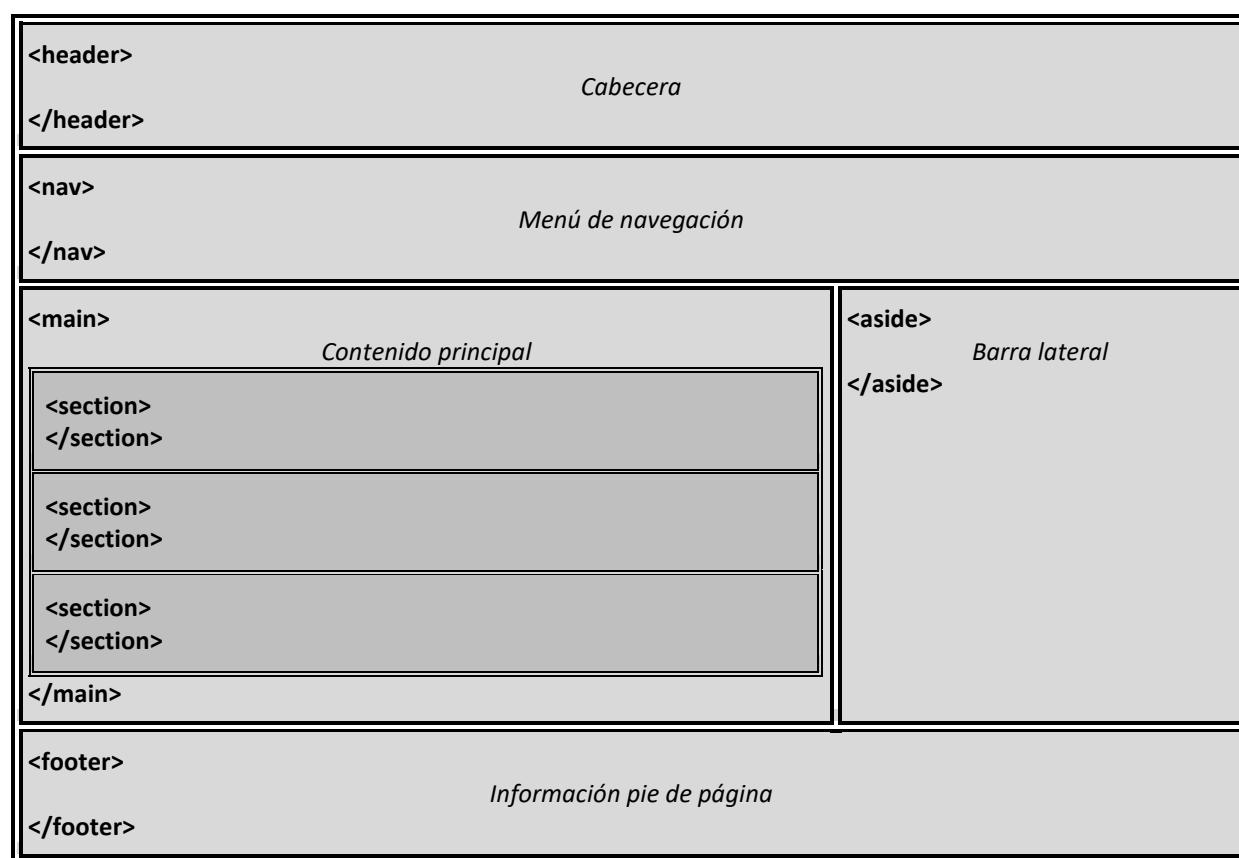
Etiqueta de bloque para especificar contenido secundario en comparación con el contenido principal, como publicidad o enlaces no tan relevantes como los del menú principal.

**MAIN**

Etiqueta de bloque para definir la región principal de los contenidos (dentro no puede haber ningún <header>, <nav>, <footer> o <aside>).

**SECTION**

Etiqueta de bloque donde se encuentra una parte de la información principal del documento, si el contenido del documento tiene diferentes partes diferenciadas.



**Generalitat  
de Catalunya**

Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)



MINISTERIO  
DE EDUCACIÓN, FORMACIÓN PROFESIONAL  
Y DEPORTES



MINISTERIO  
DE TRABAJO  
Y ECONOMÍA SOCIAL

SERVICIO PÚBLICO  
DE EMPLEO ESTATAL  
**SEPE**  
SERVICIO PÚBLICO  
D'OCUPACIÓ ESTATAL

## DEFINICIÓN DE CONTENIDOS

Cuando estructuramos el contenido principal, tenemos disponibles una serie de etiquetas para definir el significado de las distintas partes. En el siguiente diagrama se muestran dentro de una etiqueta `<section>`, pero también pueden ir a otras regiones, como el `<footer>`, y hace que su significado cambie.

### ARTICLE

Etiqueta de bloque para definir un contenido independiente.

### DETAILS

Etiqueta de bloque para definir el contenido dentro de un artículo.

### SUMMARY

Etiqueta de bloque para definir el resumen del contenido dentro de un artículo.

### FIGURE

Etiqueta de bloque para insertar una imagen.

### FIGCAPTION

Etiqueta de bloque para poner una explicación de la imagen en `<figure>`.

### DETAILS

Etiqueta de bloque para definir un contenido que por defecto quedará oculto si no ponemos el atributo `open`.

### SUMMARY

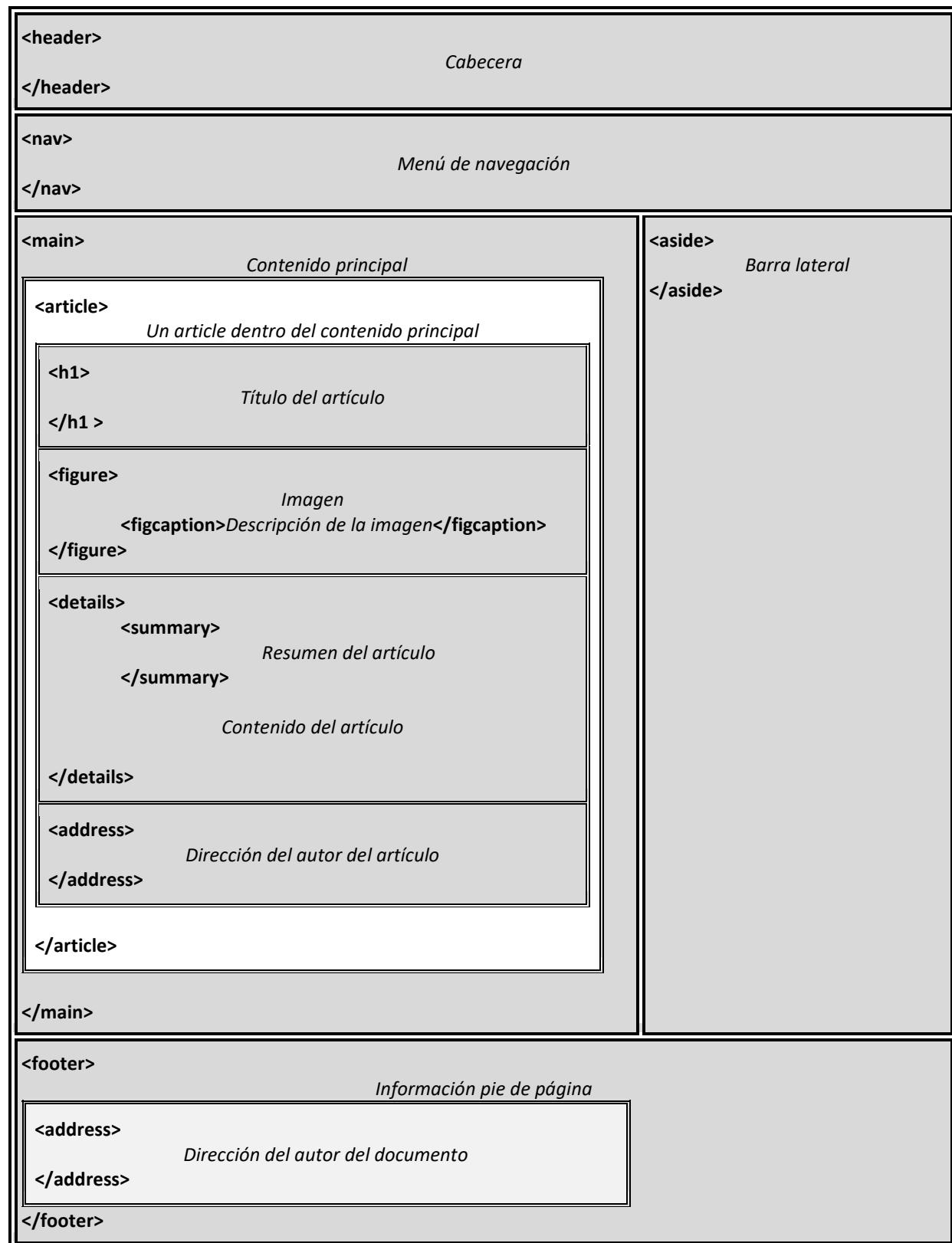
Etiqueta de bloque para definir el título o resumen del contenido de `<details>` y permite desplegar su contenido.

### ADDRESS

Etiqueta de bloque para especificar la información de contacto del autor. Dependiendo de dónde esté, puede definir la información de contacto del autor de un artículo o de todo el documento.

### DIV

Etiqueta de bloque genérica por cualquier región del documento, pero sin ningún significado concreto. Sirve para agrupar elementos y dar estilos visuales con CSS de forma conjunta.



**Generalitat  
de Catalunya**

Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)



SERVICIO PÚBLICO  
DE EMPLEO ESTATAL  
**SEPE**  
SERVICIO PÚBLICO  
D'OCUPACIÓ ESTATAL

```
<!DOCTYPE html>
<html lang="es">
    <head>
        <title>Titulo del documento</title>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <meta name="description" content="documento de ejemplo de codigo
HTML 5">
        <meta name="keywords" content="curso, html, css">
    </head>
    <body>
        <header>
            Cabecera
        </header>
        <nav>
            Menú de navegación
        </nav>
        <main>
            Contenido principal
            <article>
                Un articulo dentro del contenido principal
                <hgroup>
                    Título del artículo
                </hgroup >
                <figure>
                    Imagen
                    <figcaption>
                        Descripción de la imagen
                    </figcaption>
                </figure>
                <details>
                    <summary>
                        Resumen del artículo
                    </summary>
                    Contenido del artículo
                </details>
                <address>
                    Dirección del autor del artículo
                </address>
            </article>
        </main>
        <aside>
            Barra lateral
        </aside>
        <footer>
            Información pie de página
            <address>
                Dirección del autor del documento
            </address>
        </footer>
    </body>
</html>
```



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

## TIPOS DE CONTENIDOS PRINCIPALES

### PARÁGRAFO

La etiqueta de bloque **<p>** sirve para definir contenidos sin ningún significado especial.

### ENCABEZAMIENTO

Las etiquetas de bloque **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>** y **<h6>** sirven para definir cuándo un texto es un título del contenido. El título puede ser de todo el documento o sólo de una parte, dependiendo de qué etiqueta de organización o definición se encuentre.

El número de la etiqueta significa la importancia de este título: la etiqueta **<h1>** es la de mayor importancia y la etiqueta **<h6>** es la de menor importancia.

### LISTAS

Una parte muy importante de los contenidos de una web son los elementos relacionados entre sí, pero distintos. Un ejemplo muy claro son los vínculos dentro de una región de menú **<nav>**:

```
<nav>
  <ul>
    <li>enlace</li>
    <li>enlace</li>
    <li>enlace</li>
  </ul>
</nav>
```

Todas las listas comienzan con una etiqueta que define el tipo de lista: **<ul>** por listas no ordenadas (el orden de los elementos no es relevante) y **<ol>** por listas ordenadas (el orden sí es importante). Con el atributo **start** definimos el número de inicio de la lista, y con el atributo **reversed** invertimos el comando.

Luego utilizamos la etiqueta de bloque **<li>** para definir los diferentes elementos dentro de la lista. Cada elemento tendrá frente a un ícono si es una lista no ordenada o un número si es ordenada:

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

Mostrará:

- elemento de lista
- elemento de lista
- elemento de lista

```
<ol>
  <li>elemento de lista</li>
  <li>elemento de lista</li>
  <li>elemento de lista</li>
</ol>
```

Mostrará:

1. elemento de lista
2. elemento de lista



### 3. elemento de lista

En caso de que queramos hacer una lista anidada, hay que tener en cuenta la regla básica que dentro de una etiqueta `<ol>` o `<ul>` sólo puede haber etiquetas `<li>`; por tanto, las listas anidadas irán dentro de un `<li>`:

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

Mostrará:

- elemento de lista
- elemento de lista
  - elemento de lista
  - elemento de lista
    - elemento de lista
    - elemento de lista
    - elemento de lista
  - elemento de lista
- elemento de lista

Otro tipo de lista es la de definición con la etiqueta de bloque `<dl>`, un tipo de lista que permite poner un concepto y su definición:

```
<dl>
    <dt>concepto 1</dt>
    <dd>definición del concepto 1</dd>

    <dt>concepto 2</dt>
    <dd>definición del concepto 2</dd>
</dl>
```

Dentro de la etiqueta `<dl>` ponemos la etiqueta de bloque `<dt>` para indicar qué concepto queremos definir, y con la etiqueta de bloque `<dd>` introducimos la definición.

### CITAS

Con la etiqueta de bloque `<blockquote>` podemos citar un contenido que no es nuestro y pertenece a una fuente externa.

```
<blockquote>
```



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

E com entre los altres insignes cavallers de gloria recordació sia stat aquell valentissim cavaller Tirant lo Blanch, del qual fa special conmemoració lo present libre, per ço, de aquell e de les sues grandissimes virtuts e cavallerias se fa singular e expressa menció individual, segons reciten les següents hystòries.

</blockquote>

## SEPARADORES

Con la etiqueta de bloque <hr> podemos separar contenidos de nuestro documento.

```
<h3>Capítulo 1</h3>
<p>Contenido del primer capítulo</p>
<hr>
<h3>Capítulo 2</h3>
<p>Contenido del segundo capítulo</p>
```

## TIPOS DE CONTENIDOS

A continuación puedes consultar etiquetas de línea y su significado. Al ser etiquetas de línea, no rompen el flujo de texto y permite que los elementos marcados no se separen del resto del texto. Además, para poder utilizarlos es necesario situarlos dentro de una etiqueta de bloque, normalmente dentro de la etiqueta <p>.

## RESALTADO DE TEXTO

Las etiquetas de resaltado permite dar mayor importancia al texto marcado:

La etiqueta <strong> refuerza la importancia del texto.

La etiqueta <em> enfatiza el texto.

La etiqueta <mark> resalta el texto.

```
<strong>texto importante</strong>
<em>texto importante</em>
<mark>texto importante</mark>
```

## CITAS DE TEXTO

Las etiquetas de citas permite indicar cuándo un texto proviene de otra fuente:

La etiqueta <q> indica que el texto es original de otra fuente; con el atributo **cite** especificamos la URL de la fuente original.

La etiqueta <cite> indica que el texto es el título de una obra.

La etiqueta <abbr> indica que el texto es siglas o un acrónimo; con el atributo **title** especificamos el texto completo.

```
<q cite="https://es.wikipedia.org/wiki/Wikipedia:Acerca_de">Wikipedia en español es la versión en español de Wikipedia, un proyecto de enciclopedia web multilingüe de contenido libre basado en un modelo de edición abierta. Wikipedia crece cada día gracias a la participación de gente de todo el mundo, siendo el mayor proyecto de recopilación de conocimiento jamás realizado en la historia de la humanidad. </q>
<cite>Mecanoscrito del segundo origen</cite>
<abbr title="Tecnologías de la Información y de la Comunicación">TIC</abbr>
```



**Generalitat  
de Catalunya**



## LÍNEA BASE

Si debemos cambiar la línea base del texto podemos utilizar las siguientes etiquetas:

La etiqueta **<sup>** permite poner el texto sobre la línea base para hacer superíndice como por ejemplo 80m<sup>2</sup>

La etiqueta **<sub>** permite poner el texto debajo de la línea base para hacer subíndice como por ejemplo H<sub>2</sub>O

```
80m<sup>2</sup>
H<sub>2</sub>O
```

## CONTROL DE CAMBIOS

Las etiquetas de control de cambios permiten marcar un texto que ha sido modificado:

La etiqueta **<s>** indica que el texto está obsoleto.

La etiqueta **<del>** indica que el texto se ha borrado.

La etiqueta **<ins>** indica que el texto es nuevo o corregido.

```
<s>texto obsoleto</s>
<del>texto borrado</del>
<ins>texto corregido</ins>
```

## SALTO DE LÍNEA

Las etiquetas de salto de línea permiten continuar un texto por debajo de la línea del texto anterior dentro de un bloque:

La etiqueta **<br>**, que no es necesario cerrar, permite dar un salto de línea dentro de un bloque.

La etiqueta **<wbr>**, que no es necesario cerrar, indica en el navegador por dónde se debe hacer el cambio de línea si el texto no cabe en una sola línea en el ancho del navegador.

```
Salto de línea<br>forzado
http://ca.wikipedia.org/wiki/<wbr>_Tecnologías_de_la_información_y_la_comunicación
```

## ENLACES

Con la etiqueta **<a>** podemos marcar un texto para darle una función de enlace, aunque es necesario añadir atributos para acabar de definir su funcionamiento:

- **href**: dirección url donde queremos que apunte el enlace.
- **title**: texto explicativo sobre el documento enlazado.
- **target**: destino del documento, donde debe cargarse el documento enlazado. Puede tener diferentes valores:
  - **\_self**: valor predeterminado, el enlace se carga en la misma pestaña del navegador.
  - **\_blank**: el enlace se carga en una pestaña nueva del navegador.
  - **\_parent**: en una estructura de marcos, el documento enlazado se muestra en el marco padre.
  - **\_top**: en una estructura de marcos, el documento enlazado se muestra en el marco superior.
  - **\_valor personalizado**: en una estructura de marcos, el documento enlazado se muestra en el marco especificado.



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

- **download:** especifica que el destino debe descargarse y como valor damos el nombre que tendrá el archivo una vez descargado.
- **rel:** relación con el enlace.
- **ping:** dirección url que activamos cuando se visita la url del atributo **href**.

Según el valor que ponemos dentro del atributo **href**, tendremos distintos tipos de enlaces:

- **Enlace relativo dentro de un sitio web:** hacemos un enlace indicando la ruta por la estructura de directorios del servidor web desde el archivo de origen hacia el de destino.
- **Enlace absoluto dentro de un sitio web:** hacemos un enlace indicando la ruta por la estructura de directorios del servidor web desde el directorio raíz hacia el de destino.
- **URI:** tipos de enlace donde marcamos la ruta completa, protocolo y dominio incluido; estos enlaces comienzan con el protocolo "**http://**" para indicar que hacemos una búsqueda por internet, no por mi estructura de directorios.
- **Enlace a una marca:** hacemos un enlace hacia una marca hecha con el atributo **id**.
- **Correo electrónico:** hacemos un vínculo hacia una dirección de correo electrónico para crear un mensaje nuevo con el protocolo "**mailto:**"; podemos añadir parámetros al correo electrónico que se genera:
  - **subject:** asunto del mensaje de correo electrónico.
  - **body:** contenido del mensaje de correo electrónico.
  - **cc:** copia del mensaje de correo electrónico.
  - **bcc:** copia oculta del mensaje de correo electrónico.

```
<a title="enlace para entrar en un directorio" href="directorio/documento.html">
enlace relativo </a>

<a title="enlace para salir de un directorio" href="../document.html">enlace
relativo </a>

<a title="enlace para ir al directorio principal" href="/document.html">enlace
absoluto</a>

<a title="enlace para ir a otro sitio web en una nueva ventana del navegador"
href="https://es.wikipedia.org/wiki/Wikipedia:Portada" target="_blank">enlace
externo </a>

<span id="marca"> texto marcado </span>
<a title="enlace para ir a una marca" href="#marca">enlace a una marca</a>

<a title="enlace para crear un mensaje de correo electrónico"
href="mailto:omar.drg@outlook.es">correo electrónico</a>

<a title="enlace para crear un mensaje de correo electrónico con parámetros"
href="mailto:omar.drg@outlook.es?subject=asunto del mensaje&body=contenido del
mensaje&cc=dirección de copia&bcc=dirección de copia oculta">correo
electrónico</a>
```

## BARRAS DE PROGRESO

Con la etiqueta **<progress>** podemos marcar un porcentaje para mostrarlo visualmente, aunque es necesario añadir atributos para acabar de definir su funcionamiento:

- **value:** valor del porcentaje.
- **max:** valor total; si es un porcentaje, el valor será 100.



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

```
<progress value="32" max="100"> 32% </progress>
<progress value="0.6"> 60% </progress>
```

Con la etiqueta **<meter>** podemos dar un valor numérico escalar dentro de un rango, aunque es necesario añadir atributos para terminar de definir su funcionamiento:

- **value**: valor del porcentaje, en un valor numérico entero o decimal.
- **max**: valor total; si es un porcentaje, el valor será 100.
- **high**: valor considerado alto; si el valor está dentro del rango definido, sale con un color predefinido.
- **low**: valor considerado bajo; si el valor está dentro del rango definido, sale con un color predefinido.
- **max**: valor máximo; si es un porcentaje, el valor será 100.
- **min**: valor mínimo; si es un porcentaje, el valor será 0.
- **optimum**: valor considerado óptimo; si el valor es igual, sale con un color predefinido.

```
<meter value="32" max="100" low="20" high="100"> 32% </meter>
<meter value="0.6" low="0.2" high="0.8" optimum="0.5"> 60% </meter>
```

Se recomienda acompañar estas etiquetas con la etiqueta **<label>** para identificar el concepto que se mide.

#### SPAN

La etiqueta **<span>** no tiene ninguna función o significado especial, pero permite marcar un texto para añadir estilos CSS.

#### CONTENIDOS MULTIMÉDIA

Tenemos disponibles una serie de etiquetas de línea para poder incluir imágenes, sonidos o vídeos en nuestros documentos web.

#### IMAGENES

La etiqueta **<img>** es una etiqueta de inclusión, no es necesario cerrarla, que permite realizar un marco donde mostrar una imagen externa. En un documento html, donde sólo hay texto, las imágenes no se insertan sino que es necesario indicar en el navegador dónde se encuentra la imagen que queremos que se muestre en este marco.

Esta etiqueta necesita una serie de atributos para definir su funcionamiento:

- **src**: ruta hacia la imagen que queremos mostrar.
- **alt**: texto explicativo que aparece si la imagen no se ha cargado correctamente.
- **title**: texto explicativo que sale si pasamos el cursor por encima de la imagen.
- **height**: altura del marco de la imagen en píxeles o en porcentaje.
- **width**: ancho del marco de la imagen en píxeles o en porcentaje.
- **usemap**: mapa de enlaces de la imagen.

```

```

La etiqueta **<picture>** es una nueva etiqueta que permite definir la imagen a cargar según el tamaño de pantalla para conseguir optimizar el formato y peso.



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

El elemento <picture> contiene dos etiquetas: una o más etiquetas <source> y una etiqueta <img>.

El navegador buscará el primer elemento <source> donde la consulta multimedia coincide con la anchura de la ventana de visualización actual (especificado en el atributo **media**) y, a continuación, mostrará la imagen adecuada (especificada en el atributo **srcset**).

El elemento <img> es necesario como último hijo del elemento <picture>, como opción alternativa si ninguna de las etiquetas de origen coincide.

```
<picture>
  <source media="(min-width: 950px)" srcset="retrato_horizontal.jpg">
  <source media="(min-width: 465px)" srcset="retrato_vertical.jpg">
  
</picture>
```

## MAPAS DE IMAGENES

Si queremos convertir una imagen en un enlace, sólo deberemos utilizar la etiqueta <a> para marcar la imagen. Pero si necesitamos que una misma imagen tenga distintos enlaces, podemos utilizar la etiqueta <map>. La etiqueta <map> es una lista de etiquetas <area> por cada una de las áreas activas de la imagen.

La etiqueta <map> tiene el atributo **name** para especificar el nombre único del mapa. La etiqueta <area> tiene los siguientes atributos:

- **shape:** tipos de forma que queremos hacer. Tenemos cuatro valores distintos:
  - **default:** define la imagen entera.
  - **rect:** define un área rectangular.
  - **circle:** define un área circular.
  - **poly:** define un área poligonal.
- **coords:** en función del tipo de área seleccionada, damos las coordenadas necesarias medidas en píxeles desde la posición 0,0 de la esquina superior izquierda de la imagen:
  - rect: **x1,y1,x2,y2**: posición x y y de la esquina superior izquierda y de la esquina inferior derecha.
  - circle: **x,y,radio**: posición x y y del centro y el radio.
  - poly: **x1,y1,x2,y2,...,xn,yn**: posición x y y de cada vértice del polígono.
- **href:** dirección url donde queremos que apunte el enlace.
- **alt:** texto explicativo sobre el documento enlazado.
- **title:** texto explicativo sobre el documento enlazado.
- **target:** destino del documento, donde se debe cargar el documento enlazado. Puede tener diferentes valores:
  - **\_self:** valor predeterminado, el enlace se carga en la misma pestaña del navegador.
  - **\_blank:** el enlace se carga en una pestaña nueva del navegador.
  - **\_parent:** en una estructura de marcos, el documento enlazado se muestra en el marco padre.
  - **\_top:** en una estructura de marcos, el documento enlazado se muestra en el marco superior.
  - **\_valor personalizado:** en una estructura de marcos, el documento enlazado se muestra en el marco especificado.
- **download:** especifica que el destino debe descargarse y como valor damos el nombre que tendrá el archivo una vez descargado.

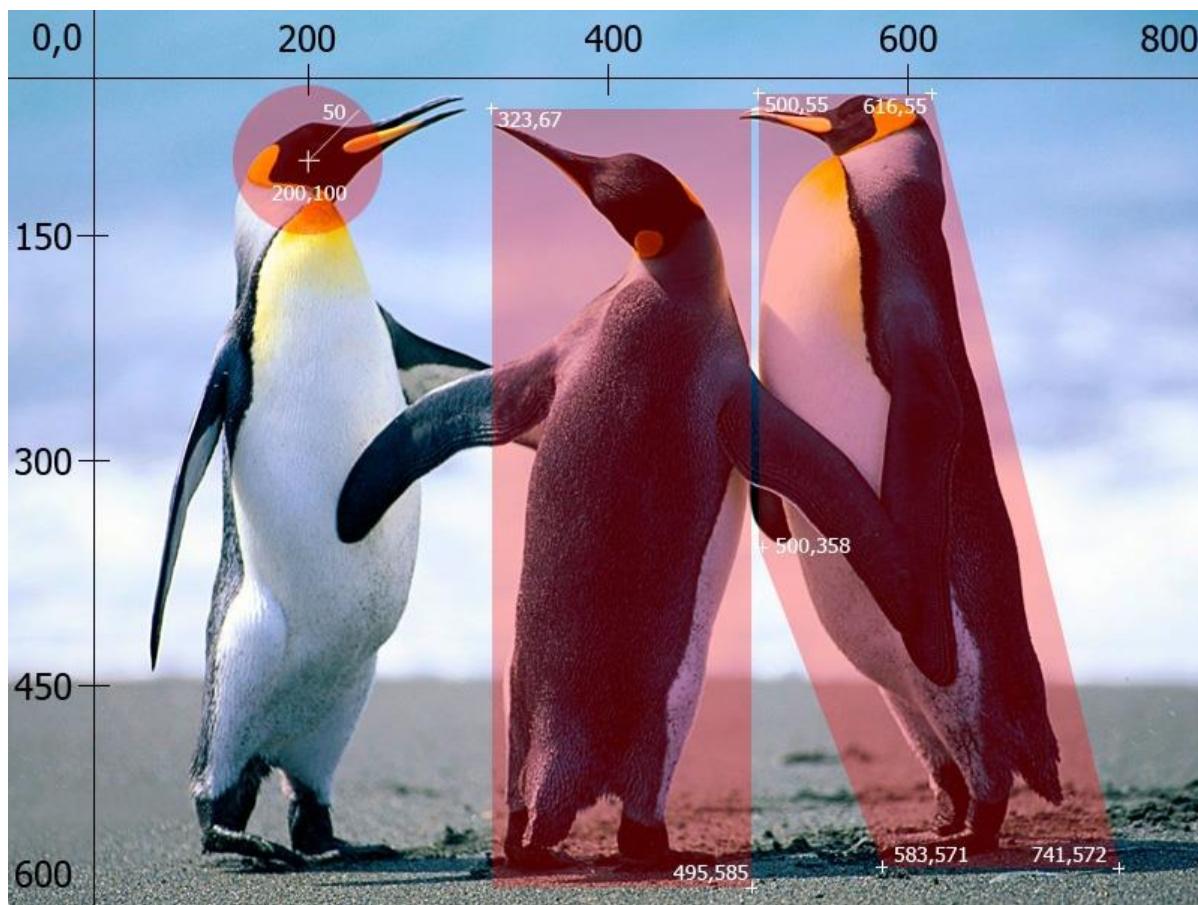


**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

Una vez creado el mapa, debemos indicar en la imagen donde queremos utilizar qué mapa debe cargar con el atributo **usemap**:



```

<map name="mapa">
<area shape="circle" coords="200,100,50" href="#" title="área1" alt="área1">
<area shape="rect" coords="323,67,495,585" href="#" title="área2" alt="área2">
<area shape="poly" coords="500,55,616,55,741,572,583,571,500,358" title="área3"
alt="área3">
</map>
```

## AUDIO

Si queremos insertar un sonido con su reproductor de control de volumen y tiempo, utilizamos la etiqueta **<audio>**. Aunque esta etiqueta no da ningún formato al texto, se abre y se cierra para poder realizar un control de versión del navegador. Sus atributos son:

- **controles**: si el navegador debe poner controles de reproducción.
- **autoplay**: si debe reproducirse automáticamente al cargarse el documento.
- **loop**: si debe reproducirse de forma infinita.
- **src**: ruta hacia el archivo que queremos reproducir. Puesto que podemos poner el audio en diferentes formatos, podemos establecer diferentes rutas con la etiqueta **<source>**

```
<audio controls autoplay loop>
  <source src="tema.ogg" type="audio/ogg">
  <source src="tema.mp3" type="audio/mpeg">
    El seu navegador no accepta audio.
</audio>
```

## VÍDEO

Si queremos insertar un vídeo con su reproductor de control de volumen y tiempo, utilizamos la etiqueta **<video>**. Aunque esta etiqueta no da ningún formato al texto, se abre y se cierra para poder realizar un control de versión del navegador. Sus atributos son:

- **controles**: si el navegador debe poner controles de reproducción.
- **autoplay**: si debe reproducirse automáticamente al cargarse el documento.
- **muted**: si debe reproducirse sin volumen (necesario para que funcione el **autoplay**)
- **loop**: si debe reproducirse de forma infinita.
- **preload**: si debe realizar una precarga previa antes de poder reproducirse.
- **src**: ruta hacia el archivo que queremos reproducir. Puesto que podemos poner el audio en diferentes formatos, podemos establecer diferentes rutas con la etiqueta **<source>**
- **poster**: ruta hacia la imagen de carátula del vídeo.
- **height**: altura del marco del vídeo en píxeles o en porcentaje.
- **width**: ancho del marco del vídeo en píxeles o en porcentaje.

```
<video controls loop preload="auto" poster="portada.jpg" >
  <source src="tema.ogg" type="video/ogg">
  <source src="tema.mp4" type="video/mp4">
    Su navegador no acepta vídeo.
</video>
```

## VENTANAS DE DIÁLOGO O EMERGENTES

La etiqueta **<dialog>** define una caja de contenido de información o diálogo. Con el atributo **open** definimos si se debe mostrar:

- **open**: fuerza mostrar la ventana de dialogo; si no está, la ventana está oculta.

```
<dialog open>Esta es una ventana de diálogo abierta</dialog>
```

Un *popover* es una ventana de notificación emergente, que funciona como elemento de control gráfico de tipo contenedor, que se desplaza sobre una ventana principal y bloquea cualquier otra interacción hasta que se seleccione.

Igual que el resto de elementos de tipo contenedor, puede contener varios elementos de contenido, como textos o elementos multimedia, o control gráfico, como casillas de verificación, botones de opción o un cuadro de lista. Como cualquier elemento de control gráfico de tipo contenedor, está destinado a agrupar elementos juntos y no debe ser extenso. En el momento que se interactúa con cualquier otro elemento o tecla ESC, este contenedor desaparece.

Para que el *popover* funcione correctamente, hace falta añadir los atributos html **popover** e **id** al contenedor emergente:



- **popover**: define un elemento como un elemento emergente, lo que significa que cuando se invoca, se colocará en la parte superior del contenido, no interferirá con la posición de otros elementos HTML.  
Puede tener diferentes valores:
  - **auto**: valor predeterminado, el contenedor se cierra al interactuar con cualquier otro elemento o pulsar la tecla ESC.
  - **manual**: el contenedor debe cerrarse interactuando con el elemento deertura u otro elemento de cierre.
- **id**: identifica al contenedor para poder controlarlo.

Además, hará falta un botón o elemento que desencadene la apertura y otro que pueda controlar el cierre con los siguientes atributos:

- **popovertarget**: con el valor del identificador del contenedor *popover*, hacemos referencia al contendor específico con el que queremos interactuar.
- **popovertargetaction**: identifica al contenedor para poder controlarlo.
  - **toggle**: valor predeterminado, el contenedor se muestra o se oculta dependiendo de su estado al pulsar el elemento.
  - **show**: el contenedor se muestra al pulsar el elemento.
  - **hide**: el contenedor se muestra al pulsar el elemento.

```
<button popovertarget="elpopover" type="button">Abre</button>
<div id="elpopover" popover>
  <button popovertarget="elpopover" popovertargetaction="hide" title="Cierra"
  type="button"> X </button>
  <h2>¡Hola mundo!</h2>
</div>
```

El propio navegador ya aplica automáticamente los estilos para que quede centrado, pero habrá que aplicar más estilos CSS para darle la apariencia necesaria.

## TABLAS

Cuando debemos maquetar una hoja de datos en formato de cuadrícula, podemos utilizar la estructura de tabla del lenguaje HTML. Para empezar hacemos una estructura **<table>** para indicar los límites de la tabla.

```
<table>
</table>
```

Dentro de la etiqueta **<table>**, introducimos las siguientes etiquetas:

- **<caption>** para poner un título en la tabla
- **<thead>**: para definir el contenido del encabezado de la tabla (generalmente los títulos de las columnas)
- **<tbody>**: para definir el contenido general de la tabla
- **<tfoot>**: para definir el contenido del pie de la tabla (generalmente resúmenes del contenido de las columnas)

```
<table>
  <caption>Título de la tabla</caption>
  <thead>
```



```
</thead>
<tbody>
</tbody>
<tfoot>
</tfoot>
</table>
```

Dentro de las etiquetas **<thead>**, **<tbody>** i **<tfoot>** introducimos etiquetas **<tr>** para indicar las diferentes filas de la tabla y, dentro de estas, introducimos etiquetas **<th>** para indicar las cabeceras de las columnas de la tabla. La etiqueta **<th>** puede tener el atributo **scope** para definir exactamente qué encabeza:

- **scope**: especifica si una celda de encabezado es un encabezado para una columna, fila o grupo de columnas o filas
  - **col**: especifica que la celda es un encabezado para una columna
  - **row**: especifica que la celda es un encabezado para una fila
  - **colgroup**: especifica que la celda es un encabezado para un grupo de columnas
  - **rowgroup**: especifica que la celda es un encabezado para un grupo de filas

```
<table>
  <caption> Título de la tabla </caption>
  <thead>
    <tr>
      <th scope="col">Cabecera 1</th>
      <th scope="col">Cabecera 2</th>
      <th scope="col">Cabecera 3</th>
    </tr>
  </thead>
  <tbody>
  </tbody>
  <tfoot>
  </tfoot>
</table>
```

Dentro de las etiquetas **<tr>**, introducimos etiquetas **<td>** para indicar los contenidos de las columnas de la tabla.

```
<table>
  <caption> Título de la tabla </caption>
  <thead>
    <tr>
      <th scope="col">Cabecera 1</th>
      <th scope="col">Cabecera 2</th>
      <th scope="col">Cabecera 3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Fila 2 - Celda 1</td>
      <td>Fila 2 - Celda 2</td>
      <td>Fila 2 - Celda 3</td>
    </tr>
    <tr>
      <td>Fila 3 - Celda 1</td>
      <td>Fila 3 - Celda 2</td>
      <td>Fila 3 - Celda 3</td>
    </tr>
    <tr>
      <td>Fila 4 - Celda 1</td>
    </tr>
  </tbody>
</table>
```



**Generalitat  
de Catalunya**

Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)



MINISTERIO  
DE EDUCACIÓN, FORMACIÓN PROFESIONAL  
Y DEPORTES



MINISTERIO  
DE TRABAJO  
Y ECONOMÍA SOCIAL



SERVICIO PÚBLICO  
DE EMPLEO ESTATAL  
**SEPE**  
SERVICIO PÚBLICO  
D'OCUPACIÓ ESTATAL

```

<td>Fila 4 - Celda 2</td>
<td>Fila 4 - Celda 3</td>
</tr>
</tbody>
<tfoot>
</tfoot>
</table>

```

Debe controlarse que cada fila tenga el mismo número de columnas. Si queremos combinar celdas, utilizamos los atributos **rowspan** y **colspan** en las etiquetas **<th>** o **<td>**, para indicar la cantidad de celdas que combinamos en vertical y horizontal respectivamente.

Fila 1 - Celda 1 + Fila 1 - Celda 2	Fila 1 - Celda 3	Fila 1 - Celda 4 + Fila 2 - Celda 4
Fila 2 - Celda 1	Fila 2 - Celda 2	Fila 2 - Celda 3

```

<table>
  <tr>
    <td colspan="2">Fila 1 - Celda 1 + Fila 1 - Celda 2</td>
    <td>Fila 1 - Celda 3</td>
    <td rowspan="2">Fila 1 - Celda 4 + Fila 2 - Celda 4</td>
  </tr>
  <tr>
    <td>Fila 2 - Celda 1</td>
    <td>Fila 2 - Celda 2</td>
    <td>Fila 2 - Celda 3</td>
  </tr>
</table>

```

## MARCOS

La etiqueta de línea **<iframe>** permite añadir contenido externo al documento (por ejemplo de otro documento) dentro del mismo documento, en un marco con un tamaño determinado. Aunque esta etiqueta no da ningún formato al texto, se abre y se cierra para poder realizar un control de versión del navegador.

Esta etiqueta necesita una serie de atributos para definir su funcionamiento:

- **src**: ruta hacia el archivo que queremos mostrar.
- **name**: nombre del marco; permite interactuar con el atributo **target** de los enlaces.
- **height**: altura del marco en píxeles o porcentaje.
- **width**: ancho del marco en píxeles o porcentaje.

```

<iframe src="documento.html" width="85%" height="350">
Su navegador no acepta marcos.
</iframe>

```

Si después queremos hacer un enlace para cambiar el documento que se carga dentro, en el **target** del enlace ponemos el valor del **name** del **<iframe>**:

```

<p><a href="documento2.html" target="marc">documento 2</a></p>
<iframe src="documento1.html" width="85%" height="350" name="marc">
Su navegador no acepta marcos.
</iframe>

```

## FORMULARIOS

Una parte muy importante de los documentos web es la funcionalidad de recabar información del visitante para poder procesarla. En un mismo documento podemos crear tantos formularios como queramos: el único requisito es que cada uno de los formularios que hacemos tengan la etiqueta de bloque **<form>** para indicar dónde comienza y dónde termina.

Esta etiqueta necesita una serie de atributos para definir su funcionamiento:

- **action:** URL del CGI a la que deben enviarse los datos. CGI es el nombre genérico que recibe cualquier plataforma que puede recoger y procesar los datos; por ejemplo: un archivo PHP que recoja los datos y los envíe por correo electrónico o los almacene en una base de datos MySQL es un CGI.
- **method:** forma de enviar los datos, en POST o en GET (por defecto).
- **enctype:** codificación de los datos que se envían. Es necesario especificar el valor "**multipart/form-data**" si queremos enviar archivos adjuntos.
- **target:** destino del documento, donde debe cargarse el documento enlazado. Puede tener diferentes valores:
  - **\_self:** valor predeterminado, el enlace se carga en la misma pestaña del navegador.
  - **\_blank:** el enlace se carga en una pestaña nueva del navegador.
  - **\_parent:** en una estructura de marcos, el documento enlazado se muestra en el marco padre.
  - **\_top:** en una estructura de marcos, el documento enlazado se muestra en el marco superior.
  - **valor personalizado:** en una estructura de marcos, el documento enlazado se muestra en el marco especificado.
- **name:** nombre del formulario para trabajar con JavaScript.
- **novalidate:** atributo para indicar si los campos del formulario deben validarse o no.
- **autocomplete:** atributo para indicar si el navegador debe dar sugerencias para completar los campos del formulario.
- **accept-charset:** codificación de los caracteres de los datos que se envían con el formulario.

```
<form action="mail.php" method="post" target="_self" accept-charset="UTF-8">  
</form>
```

## ORGANIZACIÓN DE LOS CAMPOS

Cuando organizamos los campos de un formulario, podemos utilizar etiquetas de bloque **<fieldset>** para crear contenedores. Estos contenedores pueden tener un título con la etiqueta **<legend>** dentro del contenedor.

```
<form action="mail.php" method="post" target="_self" accept-charset="UTF-8">  
  <fieldset>  
    <legend>Datos personales</legend>  
  </fieldset>  
</form>
```

Cada campo de formulario necesita que un texto le acompañe para indicar al visitante qué dato debe poner. Estos textos se pueden colocar dentro de una etiqueta de línea **<label>** para poder identificar este texto como una etiqueta del campo. Si el campo no va dentro de este **<label>**, es necesario añadir el atributo **for** para relacionar el campo y la etiqueta.

```
<form action="mail.php" method="post" target="_self" accept-charset="UTF-8">
```



**Generalitat  
de Catalunya**



MINISTERIO  
DE EDUCACIÓN, FORMACIÓN PROFESIONAL  
Y DEPORTES



MINISTERIO  
DE TRABAJO  
Y ECONOMÍA SOCIAL



SERVICIO PÚBLICO  
DE EMPLEO ESTATAL  
**SEPE**  
SERVICIO PÚBLICO  
D'OCUPACIÓ ESTATAL

```

<fieldset>
    <legend>Datos personales</legend>
    <p>
        <label for="nombre">Nombre</label>
        <input type="text" id="nombre" name="Nombre">
    </p>
</fieldset>
</form>

```

## TIPOS DE CAMPOS

### INPUT

Cuando queremos pedir un dato en un formulario utilizamos generalmente la etiqueta de línea **<input>**. Esta etiqueta inserta un elemento gráfico que indica al visitante que puede escribir, por tanto, no se cierra.

Esta etiqueta necesita una serie de atributos para definir su funcionamiento:

- **type:** tipos de información que queremos pedir; en función del tipo el navegador mostrará un elemento gráfico diferente o validará la información adecuadamente. Los distintos tipos son:
  - **button:** define un botón sin acción (se utiliza con JavaScript).
  - **checkbox:** define una casilla de verificación para que el visitante acepte una pregunta.
  - **color:** define un selector de color.
  - **date:** define un selector de fecha.
  - **datetime-local:** define un selector de fecha y hora.
  - **email:** define que el contenido es una dirección de correo electrónico.
  - **file:** define un selector de archivos.
  - **hidden:** define un campo oculto donde almacenar información no visible por el usuario.
  - **image:** define una imagen como botón de enviar el formulario.
  - **month:** define un selector de mes y año.
  - **number:** define un campo para introducir un número.
  - **password:** define un campo donde el texto introducido queda oculto.
  - **radio:** define un botón circular para escoger una opción entre varias expuestas.
  - **range:** define un control de desplazamiento para elegir un valor.
  - **reset:** define un botón de restablecer el formulario a sus valores predefinidos.
  - **search:** define un campo para realizar una búsqueda.
  - **submit:** define un botón para enviar el formulario.
  - **tel:** define un campo para introducir un número de teléfono.
  - **text:** valor por defecto; define un campo para introducir un texto.
  - **time:** define un campo para introducir una hora.
  - **url:** define un campo para introducir una dirección URL.
  - **week:** define un selector de semana y año.
- **name:** nombre del campo, necesario para después procesar esta información por parte del cgi.
- **value:** valor predeterminado del campo.
- **pattern:** determina el formato que debe tener el contenido del campo. Se introduce con una expresión regular en JavaScript ([http://www.w3schools.com/js/js\\_regexp.asp](http://www.w3schools.com/js/js_regexp.asp)).
- **placeholder:** texto que sale dentro del campo hasta que el usuario introduce un valor.
- **maxlength:** define la longitud máxima de caracteres en el campo.
- **autocomplete (on/off):** para indicar si el navegador debe dar opciones de auto completado o no.



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

- **autofocus:** para indicar si el foco del navegador (y el cursor) debe empezar en este campo al cargar el documento.
- **required:** para indicar si el navegador debe validar si se ha llenado este campo antes de enviar el formulario.
- **multiple:** define si en el campo se pueden dar más de un valor.
- **readonly:** para especificar en el navegador que el campo es sólo de lectura.
- **disabled:** para especificar en el navegador que el campo está inhabilitado.
- **list:** para especificar el <datalist> que contiene los posibles valores predefinidos.
- **checked:** para especificar si un elemento **type="checkbox"** o **type="radio"** ha de estar marcado.
- **max:** para especificar el valor máximo en un campo numérico.
- **min:** para especificar el valor mínimo en un campo numérico.
- **step:** para especificar los intervalos en un campo numérico.
- **size:** define el ancho en caracteres del campo.
- **form:** define a qué formulario pertenece el campo si éste está fuera de una etiqueta <form>.
- **tabindex:** atributo global para especificar el orden de navegación con el tabulador.
- **id:** atributo global para identificar el campo.

#### TEXTAREA

La etiqueta <textarea> permite añadir un área de texto para poder escribir libremente un contenido extenso. Esta etiqueta necesita una serie de atributos para definir su funcionamiento:

- **autofoco:** para indicar si el foco del navegador (y el cursor) debe empezar en este campo al cargar el documento.
- **cols:** para especificar el ancho del campo en caracteres.
- **disabled:** para especificar en el navegador que el campo está inhabilitado.
- **form:** define a qué formulario pertenece el campo si éste está fuera de una etiqueta <form>.
- **maxlength:** define la longitud máxima de caracteres en el campo.
- **name:** nombre del campo, necesario para después procesar esta información por parte del CGI.
- **placeholder:** texto que aparece dentro del campo hasta que el usuario introduce un valor.
- **readonly:** para especificar en el navegador que el campo es sólo de lectura.
- **required:** para indicar si el navegador debe validar si se ha llenado este campo antes de enviar el formulario.
- **rows:** para especificar la altura en cantidad de líneas visibles.
- **wrap (soft/hard):** para especificar si el texto tiene que ajustarse al ancho del campo o no.

#### DATALIST

La etiqueta <datalist> permite añadir una lista de opciones de respuesta por un elemento <input>.

#### SELECT

La etiqueta <select> permite realizar un desplegable de opciones de respuesta. Esta etiqueta necesita una serie de atributos para definir su funcionamiento:

- **autofocus:** para indicar si el foco del navegador (y el cursor) debe empezar en este campo al cargar el documento.
- **disabled:** para especificar en el navegador que el campo está inhabilitado.
- **form:** define a qué formulario pertenece el campo si éste está fuera de una etiqueta <form>.



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

- **multiple:** define si en el campo se puede seleccionar más de un valor.
- **name:** nombre del campo, necesario para después procesar esta información por parte del CGI.
- **required:** para indicar si el navegador debe validar si se ha cumplimentado este campo antes de enviar el formulario.
- **size:** define la cantidad de valores visibles.

#### OPTION

La etiqueta **<option>** define las opciones de respuesta dentro de una etiqueta **<datalist>** o **<select>**. Esta etiqueta necesita una serie de atributos para definir su funcionamiento:

- **disabled:** para especificar en el navegador que la opción está inhabilitada.
- **selected:** defineix que la opción es el valor predeterminado del listado.
- **value:** define el valor de la opción.

```
<input type="text" name="códigos" list="lenguajes">
<datalist id="lenguajes">
    <option value="HTML">
    <option value="CSS">
    <option value="JS">
    <option value="PHP">
</datalist>

<select name="códigos2">
    <option value="HTML">HTML</option>
    <option value="CSS">CSS</option>
    <option value="JS">JS</option>
    <option value="PHP">PHP</option>
</select>
```

#### OPTGROUP

La etiqueta **<optgroup>** permite agrupar etiquetas **<option>** dentro de una etiqueta **<select>** bajo un título común. Esta etiqueta necesita una serie de atributos para definir su funcionamiento:

- **disabled:** para especificar en el navegador que el grupo está inhabilitado.
- **label:** define el título del grupo.

```
<select name="códigos2">
    <optgroup label="Navegador">
        <option value="HTML">HTML</option>
        <option value="CSS">CSS</option>
        <option value="JS">JS</option>
    </optgroup>
    <optgroup label="Servidor">
        <option value="PHP">PHP</option>
    </optgroup>
</select>
```

#### ENVÍO DEL FORMULARIO

Para el envío de datos de un formulario a una web es necesario algún tipo de aplicación que sea capaz de recoger estos datos y generar un mail. Este tipo de aplicación recibe el nombre genérico de CGI. Dependiendo del tipo de alojamiento, el envío debe programarse con un lenguaje de servidor o puede que el mismo alojamiento cuente con una aplicación ya preparada.



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

Actualmente, hay muchas soluciones en línea que permiten generar los formularios con editores visuales y almacenar o gestionar los datos recogidos, pero normalmente requieren una suscripción y el pago de una tarifa según el volumen de los datos recogidos.

Por lo tanto, lo más habitual en los proyectos web es utilizar los recursos del mismo alojamiento para gestionar los datos del formulario: el servicio de envío de correos electrónicos **SMTP** y el lenguaje de servidor **PHP**. El PHP puede encargarse de almacenar la información en una base de datos o enviarla como un correo electrónico a una dirección especificada, a pesar de que, en muchos servidores web, el envío hará falta autenticarlo con una dirección de correo del mismo dominio, por ejemplo *no-reply@domini.cat*, usando el servicio SMTP del mismo servidor.

En el atributo **action** del formulario especificaremos la ruta hacia el archivo PHP que procesará los datos, y opcionalmente podemos añadir campos ocultos con los datos necesarios para procesar los datos.

```
<form method="post" action="/mail/envio.php ">  
<input type="hidden" name="destinatario" value="mi-email@dominio.es">  
...  
</form>
```

Después habrá que tener en cuenta cómo damos información al visitante sobre el resultado del envío, con una pantalla o aviso con el mensaje de si el envío se ha realizado correctamente o no.

## ATRIBUTOS GLOBALES

Cada etiqueta que hemos visto tenía unos atributos propios, pero existe un conjunto de atributos globales para todas las etiquetas:

- **accesskey**: define un atajo de teclado para activar o poner el foco sobre el elemento.
- **class**: define una o varias clases de estilos CSS.
- **contenteditable (true/false)**: define si el contenido es editable por el visitante.
- **contextmenu**: define el menú contextual personalizado que aparece al activar el botón derecho del ratón. Los menús contextuales se realizan con las etiquetas **<menu>** i **<menuitem>**.
- **data-\***: define una información privada personalizada, utilizada por aplicaciones.
- **dir (ltr/rtl/auto)**: define la dirección de lectura del texto.
- **draggable (true/false/auto)**: define si el elemento se puede arrastrar.
- **dropzone (copy/move/link)**: define que debe hacerse con el elemento arrastrado una vez se ha soltado.
- **hidden**: define que el elemento es irrelevante y no se muestra al usuario.
- **id**: define un identificador único por la etiqueta.
- **lang**: define el idioma del contenido.
- **spellcheck (true/false)**: define si el navegador debe verificar la corrección ortográfica y gramatical del contenido.
- **style**: define un estilo CSS online.
- **tabindex**: define el comando de tabulación para movernos entre los elementos.
- **title**: define información extra sobre el elemento.
- **translate (yes/no)**: define si el contenido puede traducirse o no.

Hay que tener en cuenta que algunos de estos atributos son nuevos de la versión HTML5 y no son soportados por todos los navegadores.



**Generalitat  
de Catalunya**

Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)



## CSS 3

El lenguaje **CSS** (acrónimo de *Cascading Style Sheets*, en castellano: "Hojas de Estilo en Cascada") es un lenguaje utilizado para describir la semántica de presentación (el aspecto y formato) de un documento escrito en código HTML. Es decir, con las hojas de estilo definimos la estética del contenido estructurado con el código HTML.

CSS es un lenguaje complementario al HTML para definir la apariencia con una sintaxis completamente distinta a la vista en el capítulo anterior. Por tanto, es necesario avisar al navegador cuando cambiamos de lenguaje.

### INTRODUCIR EL CSS

Las tres formas de introducir el nuevo código son:

1. **Estilos online:** todas las etiquetas HTML admiten el atributo **style**, introduciendo las propiedades CSS y sus valores dentro de las comillas para definir los estilos de esa etiqueta. Ejemplo:

```
<h1 style="font-size:18px; color:#f00;">text</h1>
```

2. **Estilos incrustados:** en el **<head>** del documento podemos abrir una región con la etiqueta **<style>** para definir estilos CSS para este documento. Ejemplo:

```
<style>  
</style>
```

En la versión HTML5, la definición de estilos es exclusivamente con CSS, por tanto, el atributo **type="text/css"** donde definimos que el lenguaje de estilos se codifica con CSS ya no es necesario.

3. **Archivo css externo:** en un documento con extensión ".css" definimos toda la información de estilos, y enlazamos los documentos ".html" con una llamada en el **<head>** del documento con la etiqueta **<link>**. Ejemplo:

```
<!DOCTYPE html>  
<html lang="es">  
  <head>  
    <title>Título del documento</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" href="css/estilos.css">  
  </head>  
  <body>  
  </body>  
</html>
```

### SELECTORES

Dentro de una región **<style>** o en un archivo de estilos externo, para aplicar los estilos debemos definir "selectores": para seleccionar a qué elementos del código HTML se aplicarán las propiedades CSS. Por ejemplo:

```
h1 { color:#ff0000; }
```



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

Primero ponemos al selector (en este caso llamamos a todos los elementos <h1>), y dentro de las claves ponemos las propiedades que queremos darle.

A continuación tienes una lista de todos los selectores por el lenguaje CSS (versiones 1, 2 y 3), con un ejemplo y una breve descripción.

Para más información puede consultar la siguiente página web:

[http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)

Selector	Ejemplo	Descripción del ejemplo
*	*	Selecciona todos los elementos
elemento	p	Selecciona todos los elementos <p>
.class	.intro	Selecciona todos los elementos con class="intro"
#id	#firstname	Selecciona el elemento con id="firstname"
Combinaciones	Ejemplo	Descripción del ejemplo
ascendente>hijo	div > p	Selecciona todos los elementos <p> que tienen el elemento <div> como padre
ascendente descendente	div p	Selecciona todos los elementos <p> dentro de los elementos <div>
hermano+ hermano	div + p	Selecciona todos los elementos <p> que vienen inmediatamente después el elemento <div>
elemento, elemento	div, p	Selecciona todos los elementos <div> y todos los elementos <p>
precedente~ hermano	p ~ ul	Selecciona todos los elementos <ul> que están precedidos de un elemento <p> en el mismo parente
Atributos	Ejemplo	Descripción del ejemplo
[attribute]	[target]	Selecciona todos los elementos con el atributo target
[attribute=value]	[target=_blank]	Selecciona todos los elementos con el atributo target="_blank"
[attribute~=value]	[title~=codi]	Selecciona todos los elementos que el atributo title contenga la palabra "código"
[attribute =value]	[lang =es]	Selecciona todos los elementos que el atributo lang empiece por la palabra "es"
[attribute^=value]	a[href^="https"]	Selecciona todos los elementos <a> que el atributo href empiece por "https" (versión css 3)
[attribute\$=value]	a[href\$=".pdf"]	Selecciona todos los elementos <a> que el atributo href finalice con la palabra ".pdf" (versión css 3)
[attribute*=value]	a[href*="codi"]	Selecciona todos los elementos <a> que el atributo href contenga la palabra "código" (versión css 3)
Pseudo-clases	Ejemplo	Descripción del ejemplo
:active	a:active	Selecciona el enlace activo
:any-link	a:anylink area:anylink	Selecciona cualquier elemento <a> o <area> con un atributo href
:auto-fill	input:autofill	Selecciona cualquier elemento <input> con su valor rellenado automáticamente por el navegador
:checked	input:checked	Selecciona todos los elementos <input> marcados
:default	input:default	Selecciona los elementos de formulario por defecto en un grupo de elementos relacionados
:dir()	:dir(ltr) :dir rtl)	Selecciona cualquier elemento con la dirección del texto especificada



Generalitat  
de Catalunya



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

:disabled	input:disabled	Selecciona cualquier <input> que esté inhabilitado
:empty	div:empty	Selecciona todos los elementos <div> que no tienen ningún contenido
:enabled	input:enabled	Selecciona todos los elementos <input> habilitados
:first	@page :first	Representa la primera página de un documento impreso (utilizado con la regla @page)
:first-child	p:first-child	Selecciona cada elemento <p> que es el primer hijo de su padre
:first-of-type	p:first-of-type	Selecciona cada elemento <p> que es el primer elemento <p> de su padre
:focus	input:focus	Selecciona el elemento <input> con el foco
:focus-visible	button:focus-visible	Selecciona el elemento que recibe el foco (se utiliza para aplicar estilos de enfoque sólo cuando se utiliza el teclado para enfocar algo, no el ratón)
:focus-within	form:focus-within label:focus-within	Coincide con un elemento si el elemento o cualquiera de sus descendientes recibe el foco
:fullscreen	:fullscreen	Selecciona cualquier elemento que esté actualmente en modo de pantalla completa
:has(selector)	h2:has(+p)	Selecciona los elementos h2 que van seguidos inmediatamente por un elemento p y aplica el estilo a h2
:hover	a:hover	Selecciona el enlace con el cursor encima
:in-range	input:in-range	Selecciona todos los elementos <input> con su valor dentro del rango especificado en sí mismo <input type="number" min="1" max="10">
:indeterminate	input:indeterminate	Selecciona cualquier elemento de formulario que se encuentre en estado indeterminado
:invalid	input:invalid	Selecciona todos los elementos <input> con un valor no válido
:is(selector)	:is(ul, ol)	Selecciona todos los elementos <ul> y <ol>
:lang(lenguaje)	p:lang(es)	Selecciona cada elemento <p> con el atributo lang="es"
:last-child	p:last-child	Selecciona cada elemento <p> que es el último hijo de su padre
:last-of-type	p:last-of-type	Selecciona cada elemento <p> que es el último elemento <p> de su padre
:left	@page :left	Representa todas las páginas de la izquierda de un documento impreso (utilizado con la regla @page)
:link	a:link	Selecciona todos los enlaces no visitados
:modal	dialog:modal	Selecciona el elemento <dialog> que se encuentra en un estado modal
:not(selector)	:not(p)	Selecciona todos los elementos que no son el elemento <p>
:nth-child(n)	p:nth-child(2)	Selecciona cada elemento <p> que es el segundo hijo de su padre; si queremos seleccionar los elementos pares, utilizamos el valor <b>even</b> , y por los impares el valor <b>odd</b>
:nth-last-child(n)	p:nth-last-child(2)	Selecciona cada elemento <p> que es el segundo elemento de su padre contando desde el último elemento hijo
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selecciona cada elemento <p> que es el segundo elemento <p> de su padre contando desde el último elemento hijo
:nth-of-type(n)	p:nth-of-type(2)	Selecciona cada elemento <p> que es el segundo elemento <p> de su padre



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

:only-of-type	p:only-of-type	Selecciona cada elemento <p> que es el único elemento <p> de su padre
:only-child	p:only-child	Selecciona cada elemento <p> que es el único elemento hijo de su padre
:optional	input:optional	Selecciona todos los elementos <input> que no son obligatorios
:out-of-range	input:out-of-range	Selecciona todos los elementos <input> con su valor fuera del rango especificado en sí mismo <input type="number" min="1" max="10">
:placeholder-shown	input:placeholder-shown	Selecciona cualquier <input> elemento que actualmente muestra texto de marcador de posición
:popover-open	[popover]:popover-open	Selecciona cualquier elemento que se encuentre en un estado emergente de visualización
:read-only	input:read-only	Selecciona todos los elementos <input> que son sólo de lectura (atributo "readonly")
:read-write	input:read-write	Selecciona todos los elementos <input> que son de lectura y escritura (sin el atributo "readonly")
:required	input:required	Selecciona todos los elementos <input> que son obligatorios
:right	@page :right	Representa todas las páginas de la derecha de un documento impreso (utilizado con la regla @page)
:root	:root	Selecciona el elemento principal del documento (normalmente el elemento <html>)
:scope	@scope :scope	Selecciona elementos que son un punto de referencia o ámbito para los selectores
:target	#tab1:target	Selecciona el elemento activo id="#tab1" tomando el enlace que contiene el nombre del ancla
:user-invalid	input:user-invalid	Selecciona cualquier elemento de formulario con un valor no válido (después de que el usuario haya interactuado con él)
:user-valid	input:user-valid	Selecciona cualquier elemento de formulario con un valor válido (después de que el usuario haya interactuado con él)
:valid	input:valid	Selecciona todos los elementos <input> con un valor válido
:visited	a:visited	Selecciona todos los enlaces visitados
:where(selector)	:where(ol, ul)	Selecciona todos los elementos <ul> y <ol>. Al contrario de :is(), no reescribe reglas previas
Pseudo-elementos	Ejemplo	Descripción del ejemplo
::after	p::after	Inserta contenido después de cada elemento <p>
::backdrop	[popover]::backdrop	Aplicar estilo al cuadro de vista detrás de un cuadro de diálogo <dialog> o un elemento emergente popover
::before	p::before	Inserta contenido antes de cada elemento <p>
::file-selector-button	::file-selector-button	Selecciona qualsevol botón de tipo <input type="file">
::first-letter	p::first-letter	Selecciona la primera letra de cada elemento <p>
::first-line	p::first-line	Selecciona la primera línea de cada elemento <p>
::grammar-error	::grammar-error	Estiliza un texto que el navegador ha marcado como gramaticalmente incorrecto
::marker	li::marker	Selecciona los marcadores de los elementos de la lista
::placeholder	input::placeholder	Estiliza el texto del marcador de posición de los elementos <input>
::selection	::selection	Selecciona la parte del elemento seleccionada por el usuario



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

::spelling-error	::spelling-error	Estiliza un texto que el navegador ha marcado como mal escrito
Selector de nidificación	Ejemplo	Descripción del ejemplo
&	a { &:hover {}}	Aplica estilos para un elemento dentro del contexto de otro elemento

## COMPATIBILIDAD CON VERSIONES ANTERIORES DE LOS NAVEGADORES

Hay que tener en cuenta que los nuevos atributos CSS 3 y etiquetas HTML 5 no son compatibles con versiones antiguas de los navegadores, principalmente por IExplorer, por tanto, es necesario introducir cambios según la versión. Una forma sencilla de hacerlo es mediante condicionales del propio código html:

```
<!--[if lte IE 8]>
    <link rel="stylesheet" type="text/css" href="ie8-and-down.css">
<![endif]-->
```

En este caso el código "lte IE 8" significa igual o inferior a IE versión 8. Otras opciones son:

- **gt:** superior que
- **gte:** igual o superior que
- **lt:** inferior que
- **lte:** igual o inferior que
- **!=:** diferente que

Dentro de la condición se pone el <link> hacia una hoja de estilos alternativa.

Otra opción es cargar un <script> de javascript que haga reconocer las nuevas etiquetas y atributos. Algunas de estas librerías son **Modernizr** (<http://modernizr.com/>) o **html5shiv** (<https://github.com/aFarkas/html5shiv>)

## COMPATIBILIDAD CON LAS NUEVAS PROPIEDADES

Hay navegadores que todavía no tienen las propiedades CSS 3 completamente incorporadas, por tanto, algunos atributos deben utilizarse con códigos condicionales según los navegadores. Estos son los códigos:

- **-webkit-** para Chrome, Safari i Opera
- **-moz-** para Firefox
- **-ms-** para Explorer
- **-o-** para Opera

En esta tabla encontrarás una referencia de atributos y compatibilidades:

[http://www.w3schools.com/cssref/css3\\_browsersupport.asp](http://www.w3schools.com/cssref/css3_browsersupport.asp)

## PROPIEDADES

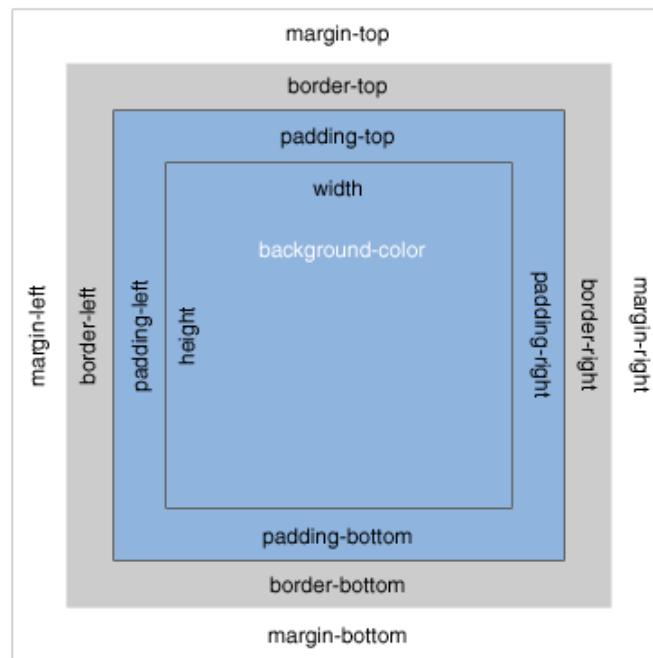
Dentro de la definición de un selector o en un atributo style en cualquier etiqueta podemos utilizar las distintas propiedades del lenguaje.

## MODELO DE CAJA

El *box model* o modelo de caja es la forma en la que el navegador interpreta las etiquetas y las muestra en pantalla. A estas etiquetas de bloque las da unos atributos y tamaños:



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)



## MARGENES

Propiedad	Descripción	Valores
<b>margin-top</b>	Tamaño del margen superior	[ <longitud>   <porcentaje>   auto   initial   inherit ]
<b>margin-right</b>	Tamaño del margen derecho	[ <longitud>   <porcentaje>   auto   initial   inherit ]
<b>margin-bottom</b>	Tamaño del margen inferior	[<longitud>   <porcentaje>   auto   initial   inherit]
<b>margin-left</b>	Tamaño del margen izquierdo	[ <longitud>   <porcentaje>   auto   initial   inherit ]
<b>margin</b>	Tamaño de los cuatro márgenes	[ <longitud>   <porcentaje>   auto   initial   inherit ] {1,4}
<b>margin-inline</b>	Tamaño de los márgenes derecho e izquierdo	[ <longitud>   <porcentaje>   auto   initial   inherit ] {1,2}
<b>margin-block</b>	Tamaño de los márgenes superior e inferior	[ <longitud>   <porcentaje>   auto   initial   inherit ] {1,2}

## MARGENES INTERIORES

Propiedad	Descripción	Valores
<b>padding-top</b>	Tamaño del margen superior	[ <longitud>   <porcentaje>   auto   initial   inherit ]
<b>padding-right</b>	Tamaño del margen derecho	[ <longitud>   <porcentaje>   auto   initial   inherit ]
<b>padding-bottom</b>	Tamaño del margen inferior	[ <longitud>   <porcentaje>   auto   initial   inherit ]
<b>padding-left</b>	Tamaño del margen izquierdo	[ <longitud>   <porcentaje>   auto   initial   inherit ]
<b>padding</b>	Tamaño de los cuatro márgenes	[ <longitud>   <porcentaje>   auto   initial   inherit ] {1,4}



**Generalitat  
de Catalunya**



MINISTERIO  
DE EDUCACIÓN, FORMACIÓN PROFESIONAL  
Y DEPORTES



MINISTERIO  
DE TRABAJO  
Y ECONOMÍA SOCIAL



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

		[ initial   inherit ] {1,4}
<b>padding-inline</b>	Tamaño de los márgenes derecho e izquierdo	[ <longitud>   <porcentaje>   auto   initial   inherit ] {1,2}
<b>padding-block</b>	Tamaño de los márgenes superior e inferior	[ <longitud>   <porcentaje>   auto   initial   inherit ] {1,2}

## BORDES

Propiedad	Descripción	Valores
<b>border-width</b>	Ancho del borde	[ thin   <b>medium</b>   thick   <longitud> ] {1,4}
<b>border-color</b>	Color del borde	[ <color>   transparente ] {1,4}
<b>border-style</b>	Estilo del borde	[ <b>none</b>   hidden   dotted   dashed   solid   double   groove   ridge   inset   outset ] {1,4}
<b>border-radius</b>	Curva del borde	[1-4 length %/1-4 length % initial inherit ]
<b>border-image-source</b>	Imagen del borde	[none <url> initial inherit ]
<b>border-image-slice</b>	Área interior del borde	[number % fill initial inherit ]
<b>border-image-width</b>	Tamaño del borde	
<b>border-image-outset</b>	Tamaño de la imagen que sobresale del borde	[<longitud>   <número>   initial   inherit ]
<b>border-image-repeat</b>	Repetición de la imagen del borde.	[ stretch   repeat   round   initial   inherit ]
<b>border-top</b>	Borde superior	[ ancho    estilo    color ]
<b>border-right</b>	Cerca	[ ancho    estilo    color ]
<b>border-bottom</b>	Borde inferior	[ ancho    estilo    color ]
<b>border-left</b>	Borde izquierdo	[ ancho    estilo    color ]
<b>border</b>	Borde de la caja	[ ancho    estilo    color ] {1,4}

## MODELO DEL FORMATO VISUAL

Propiedad	Descripción	Valores
<b>display</b>	Comportamiento del contenedor	[ inline   block   inline-block   flex   inline-flex   grid   inline-grid   list-item   run-in   table   inline-table   table-row-group   table-header-group   table-footer-group   table-row   table-column-group   table-column   table-cell   table-caption   none   initial   inherit ]
<b>position</b>	Tipo de posicionamiento	[ static   relative   absolute   fixed   sticky ]
<b>top</b> <b>right</b> <b>bottom</b> <b>left</b>	Desplazamiento de la caja	[ <longitud>   <porcentaje>   auto   initial   inherit ]
<b>inset</b>	Desplazamiento de la caja en las cuatro posiciones	[ <longitud>   <percentatge>   auto   initial   inherit ] {1,4}
<b>float</b>	Posicionamiento flotante	[ left   right   <b>none</b> ]
<b>clear</b>	Control de cajas junto a los floats	[ <b>none</b>   left   right   both ]
<b>z-index</b>	Solapamiento de niveles de capas	[ auto   <enter_amb_signe> ]



**Generalitat  
de Catalunya**



SERVICIO PÚBLICO  
DE EMPLEO ESTATAL  
**SEPE**  
SERVICIO PÚBLICO  
D'OCUPACIÓ ESTATAL

Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

<b>box-sizing</b>	Indica en el navegador que debe incluir el height y width	[ <b>content-box</b>   border-box   initial   inherit ]
<b>box-shadow</b>	Sombra de la caja	[none   h-shadow v-shadow blur spread color   inset   initial   inherit ]

Con la propiedad **display** podemos cambiar el comportamiento habitual de las etiquetas html para poder cambiar así como se comportan:

- **inline**: las etiquetas no rompen el flujo de texto.
- **block**: las etiquetas rompen el flujo de texto y pueden contener otras etiquetas. Además, pueden tener atributos de ancho y altura.
- **inline-block**: las etiquetas no rompen el flujo de texto, pero pueden tener atributos de ancho y altura.
- **flex**: permite distribuir los elementos de forma flexible y se puede controlar su distribución en el eje vertical u horizontal.
- **grid**: permite distribuir los elementos de forma de cuadrícula y se puede controlar su distribución en el eje vertical y horizontal.

Con la propiedad **position** podemos establecer cómo se posicionan:

- **static**: posición por defecto del navegador: aparecen en el orden del mismo código.
- **relative**: el elemento se posiciona respecto a su posición normal.
- **absolute**: el elemento se posiciona respecto a su antecesor posicionado (no static).
- **fixed**: el elemento se posiciona respecto a la ventana del navegador.
- **sticky**: el elemento se posiciona **relative** o **fixed** dependiendo de la posición del desplazamiento del usuario.

#### DETALLES DEL MODELO DEL FORMATO VISUAL

Propiedad	Descripción	Valores
<b>width</b>	Ancho	[ <longitud>   <porcentaje>   <b>auto</b> ]
<b>min-width</b>	Ancho mínimo	[ <longitud>   <porcentaje> ]
<b>max-width</b>	Ancho máximo	[ <longitud>   <porcentaje>   <b>none</b> ]
<b>height</b>	Altura	[ <longitud>   <porcentaje>   <b>auto</b> ]
<b>min-height</b>	Altura mínima	[ <longitud>   <porcentaje> ]
<b>max-height</b>	Altura máxima	[ <longitud>   <porcentaje>   <b>none</b> ]
<b>line-height</b>	Altura de la línea de texto	[ <b>normal</b>   <número>   <longitud>   <porcentaje> ]
<b>vertical-align</b>	Alineación vertical del texto	[ <b>baseline</b>   sub   super   top   text-top   middle   bottom   text-bottom   < porcentaje >   <longitud> ]
<b>aspect-ratio</b>	Permite definir la relación entre la anchura y la altura de un elemento	[ número/número   initial   inherit ]



## EFFECTOS VISUALES

Propiedad	Descripción	Valores
<b>overflow</b>	Comportamiento del contenido si se desborda en la caja	[ <b>visible</b>   hidden   scroll   auto ]
<b>clip</b>	Especifica la región visible del elemento	[ rect (<longitud>, <longitud>, <longitud>, <longitud>)   auto ]
<b>visibility</b>	Visibilidad de las cajas	[ <b>visible</b>   hidden   collapse ]

## CONTENIDO GENERADO, NUMERACIÓN AUTOMÀTICA Y LISTAS

Propiedad	Descripción	Valores
<b>content</b>	Agregar contenido con :after y :before	[ <b>normal</b>   none   [ <text>   <uri>   <contador>   attr(X)   open-quote   close-quote   no-open-quote   no-close-quote]+ ]
<b>quotes</b>	Especifica las marcas para indicar las citas	[ [ <text> <text> ]+   none ]
<b>counter-reset</b>	Inicializa un contador	[ [ <identificador> <enter>? ]+   none ]
<b>counter-increment</b>	Incrementa un contador	[ [ <identificador> <enter>? ]+   none ]
<b>list-style-type</b>	Estilo aplicable a los marcadores visuales de las listas	[ <b>disc</b>   circle   square   decimal   decimal-leading-zero   lower-roman   upper-roman   lower-greek   lower-latin   upper-latin   armenian   georgian   lower-alpha   upper-alpha   none ]
<b>list-style-image</b>	Imagen aplicable a los elementos de las listas	[ url("http://...")   none ]
<b>list-style-position</b>	Posición dentro de la lista de los elementos marcadores de las listas	[ <b>inside</b>   <b>outside</b> ]
<b>list-style</b>	Permite establecer el estilo de la lista, la imagen y/o la posición	[ <list-style-type>    <list-style-position>    <list-style-image> ]

## COLORES Y FONDO

Propiedad	Descripción	Valores
<b>color</b>	Color del primer plano	<color>
<b>background-color</b>	Color de fondo	[ <color>   transparente   initial   inherit ]
<b>background-image</b>	Imagen de fondo	[ url(...) , url(...)   none   initial   inherit ]
<b>background-repeat</b>	Repetición de la imagen de fondo	[ repeat   repeat-x   repeat-y   no-repeat ]
<b>background-attachment</b>	Desplazamiento de la imagen de fondo	[ scroll   fixed   local   initial   inherit ]
<b>background-position</b>	Posición de la imagen de fondo	[ [ <porcentaje>   <longitud>   left   center   right ] [ <porcentaje>   <longitud>   top   center   bottom]? ]   [ [ left   center   right ]   [ top   center   bottom ]   initial   inherit ]



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

<b>background-clip</b>	Área para cubrir del fondo	[ [ <b>border-box</b>  padding-box content-box ] initial inherit ]
<b>background-origin</b>	Posición inicial de la imagen de fondo	[ [ <b>padding-box</b>  border-box content-box ] initial inherit ]
<b>background-size</b>	Tamaño de la imagen de fondo	[ <b>auto</b>   <longitud>   <porcentaje>   cover   contain   initial   inherit ]
<b>background</b>	Propiedades individuales relacionadas con el fondo	[ [color   position/size   repeat   origin   clip   attachment   image]   initial   inherit ]

Para codificar los colores tenemos 5 formas:

- Código hexadecimal: #RRGGBB
- Código hexadecimal simplificado: #RGB
- Código decimal: `rgb( [0-255] , [0-255] , [0-255] )`
- Código por porcentajes: `rgb( [0-100]% , [0-100]% , [0-100]% )`
- Código decimal con opacidad: `rgba( [0-255] , [0-255] , [0-255] , [0-1] )`

Para colorear con degradados, utilizamos dentro de la propiedad **background** los valores:

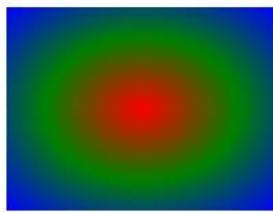
- **linear-gradient** ( [down|up|left|right|Xdeg] , [top-color [porcentaje%]] , [bottom-color [porcentaje%]] )

```
#grad1 {
    height: 200px;
    background: -webkit-linear-gradient(red, blue); /* Safari 5.1-6.0 */
    background: -o-linear-gradient(red, blue); /* Opera 11.1-12.0 */
    background: -moz-linear-gradient(red, blue); /* Firefox 3.6-15 */
    background: linear-gradient(red, blue);
}
```



- **repeating-linear-gradient** ([dirección] , [top-color [porcentaje%]] , [bottom-color [porcentaje%]] )
- **radial-gradient** ( [ellipse|circle] , [color-inicial [porcentaje%]] , [color-final [porcentaje%]] )

```
#grad1 {
    height: 150px;
    width: 200px;
    background: -webkit-radial-gradient(red, green, blue); /*Safari 5.1-6.0*/
    background: -o-radial-gradient(red, green, blue); /*Opera 11.6- 12.0*/
    background: -moz-radial-gradient(red, green, blue); /*Firefox 3.6-15*/
    background: radial-gradient(red, green, blue);
}
```



- **repeating-radial-gradient** ( [forma] , [color-inicial [porcentaje%]] , [color-final [porcentaje%]] )

Estas formas de hacer degradados dependen del navegador. Una buena herramienta online para generarlos es <http://www.colorzilla.com/gradient-editor/>

## FUENTES

Propiedad	Descripción	Valores
<b>font-family</b>	Familias de fuentes	[ [ <nombre-familia>   <familia-genérica> ] , <nombre-familia>   <familia-genérica> ]* ]
<b>font-style</b>	Estilo de la fuente	[ <b>normal</b>   italic   oblique ]
<b>font-variant</b>	Convierte las minúsculas a mayúsculas pero mantienen un tamaño inferior a las mayúsculas	[ <b>normal</b>   small-caps ]
<b>font-weight</b>	Intensidad de la fuente	[ <b>normal</b>   bold   bolder   lighter   100   200   300   <b>400</b>   500   600   700   800   900 ]
<b>font-size</b>	Tamaño de la fuente	[ [ xx-small   x-small   small   <b>medium</b>   large   x-large   xx-large ]   [larger   smaller]   <longitud>   < porcentaje > ]
<b>font</b>	Atajo para establecer el resto de propiedades sobre las fuentes a la vez	[ [ [ <font-style>    <font-variant>    <font-weight> ]? <font-size> [ / <line-height> ]? <font-family> ]   caption   icon   menu   message-box   small-caption   status-bar ]

Las fuentes que utilizamos deben ser fuentes instaladas en el sistema del usuario. Si queremos utilizar una fuente propia, primero debemos incorporarla al navegador con la propiedad **@font-face**:

```
@font-face {
    font-family: 'corpo';
    src: url('fonts/corpo.eot');
    src: url('fonts/corpo.eot?#iefix') format('embedded-opentype'),
         url('fonts/corpo.ttf') format('truetype');
    font-weight: normal;
    font-style: normal;
}
```

Con la propiedad **font-family** establecemos el nombre de la fuente que utilizaremos después, y la propiedad **src** establece la ruta hacia el archivo de la fuente que queremos utilizar. Para evitar problemas entre versiones de navegador, utilizamos el formato "eot" para Explorer y "ttf/otf" para el resto.

Otra herramienta online para utilizar fuentes diferentes es <https://fonts.google.com/>

## TEXTO

Propiedad	Descripción	Valores
<b>text-indent</b>	Desplazamiento de la primera línea del texto	[ <longitud>   <porcentaje> ]
<b>text-align</b>	Alineamiento del texto	[ left   right   center   justify ]
<b>text-decoration</b>	Efectos de subrayado, rayado, parpadeo	[ none   [ underline    overline    line-through    blink ] ]
<b>letter-spacing</b>	Espacio entre caracteres	[ normal   <longitud> ]
<b>word-spacing</b>	Espacio entre palabras	[ normal   <longitud> ]
<b>text-transform</b>	Transformaciones del texto en mayúsculas/minúsculas	[ capitalize   uppercase   lowercase   none ]
<b>white-space</b>	Comportamiento de los espacios dentro de los elementos	[ normal   pre   nowrap   pre-wrap   pre-line ]

## TABLAS

Propiedad	Descripción	Valores
<b>caption-side</b>	Posición del título de respeto a la tabla	[ top   bottom ]
<b>table-layout</b>	Control del algoritmo usado para el formato de las celdas, filas y columnas	[ auto   fixed ]
<b>border-collapse</b>	Selección del modelo de los bordes	[ collapse   separate ]
<b>border-spacing</b>	Espaciado entre los bordes de celdas adyacentes	<longitud><longitud>?
<b>empty-cells</b>	Visibilidad de los bordes de celdas sin contenido	[ show   hide ]

## INTERFICIE DE USUARIO

Propiedad	Descripción	Valores
<b>cursor</b>	Especifica el cursor	[ [ http://.../cursor.ico,]* [ auto   crosshair   default   pointer   move   e-resize   ne-resize   nw-resize   n-resize   se-resize   sw-resize   s-resize   w-resize   text   wait   help   progress ] ]
<b>outline-width</b>	Ancho de la línea exterior	<border-width>
<b>outline-style</b>	Estilo de la línea exterior	<border-style>
<b>outline-color</b>	Color de la línea exterior	[ <color>   invert ]
<b>outline</b>	Propiedades individuales de la línea exterior	[ <outline-color>    <outline-style>    <outline-width> ]
<b>speak-header</b>	(Aural) Indica si las cabeceras de la mesa se leen antes de cada celda	[ once   always ]

## MEDIA QUERIES

El concepto de *media queries* lo podemos definir en castellano como "consulta a medios", y esto es lo que realmente trabajamos con esta funcionalidad de CSS3. Hoy en día vemos la gran evolución de la comunicación, especialmente la web: últimamente se ha incrementado de forma considerable la manera en que personas



**Generalitat  
de Catalunya**



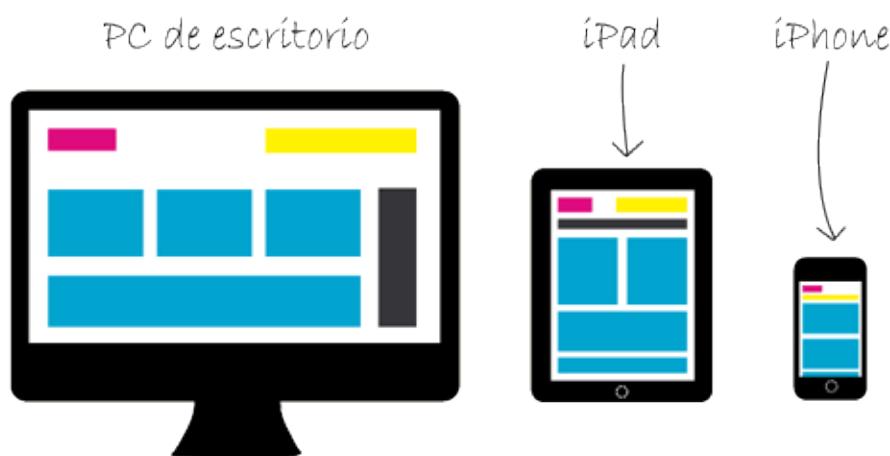
Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

acceden a la web. Se puede acceder desde televisores, pantallas de ordenadores, teléfonos móviles, tabletas e incluso consolas de videojuegos. Lo que se convierte en un reto para el diseñador o desarrollador web, hacer que la web se visualice bien en los distintos dispositivos y distintas resoluciones de pantalla, desde un teléfono inteligente hasta un televisor de gran resolución.

Esto podemos resolverlo gracias a los *media queries*, que nos permiten personalizar los estilos basados en las características del dispositivo del visitante de nuestra web. Soporte de media queries en los distintos navegadores:



Visualización de un diseño web utilizando *media queries*:

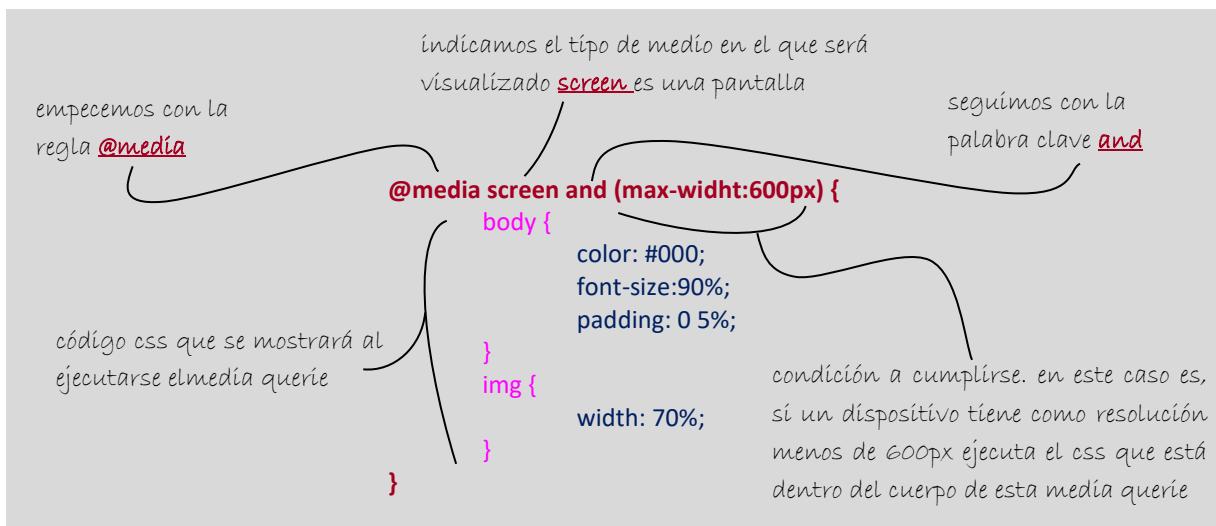


Pero veamos cómo podemos añadir estos estilos condicionales y la consulta querías que si se da la condición mostrará estos estilos.

Una de las formas para hacer esto es añadiendo la regla dentro de nuestro css principal o también creando una hoja independiente y después adjuntarla. Veamos la regla condicional de que en caso de cumplirse ejecutara ciertos estilos css:

```
@media screen and (max-width:600px) {
    body {
        color: #000;
        font-size:90%;
        padding: 0 5%;
    }
}
```

Analizar la sintaxis de la *media queries* de css:



Lo que en este caso debemos tener presente es la función o condicional que al cumplirse ejecuta el css preparado para el dispositivo en cuestión.

En el caso anterior era **max-width:600px**, es decir cualquier persona que abra la web desde un dispositivo cuya resolución sea igual o menor de 600px automáticamente ejecutará el css que está en su cuerpo. Ahora, hay muchas más opciones de condicionales que funcionan con media queries, y es lo que denominamos **media features**, miramos cuáles podemos utilizar para ejecutar css condicionales.

#### 1 - width

```
@media screen and (min-width: 400px) and (max-width: 700px) { ... }
```

*Media querie* que va a funcionar si la web se abre en un dispositivo entre los rangos de 400 y 700px. max-width es el ancho máximo y min-width el ancho mínimo.

#### 2 - height

```
@media screen and (height: 700px) { ... }
```

Con esta función especificamos la altura de visualización del dispositivo, acepta los prefijos min-/max-.

#### 3 - device-width

```
@media screen and (device-width: 800px) { ... }
```

Especificamos el ancho del área de representación, la anchura de la hoja web, acepta los prefijos min-/max-.

#### 4 - device-height

```
@media screen and (device-height:600px) { ... }
```

Especificamos la altura del área de representación, la altura de la hoja web, acepta los prefijos min-/max-.



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

## 5 - orientation

```
@media all and (orientation:portrait) {...}
@media all and (orientation:landscape) {...}
```

Especificamos la Orientación del dispositivo, podemos tener como valores ***portrait*** o ***landscape***.

La orientación es ***portrait*** cuando el *media feature width* es igual o mayor a la *media feature height*, de lo contrario la orientación es ***landscape***.

## 6 - aspect-ratio

```
@media screen and (aspect-ratio: 1280/720)
```

Especificamos la relación del dispositivo entre el *media feature width* y el *media feature height*, acepta los prefijos min-/max-.

## 7 - device-aspect-ratio

```
@media screen and (device-aspect-ratio: 1280/720) {...}
```

Especificamos la relación del área de representación entre el *media feature width* y el *media feature height*, acepta los prefijos min-/max-.

## 8 - color

```
@media all and (min-color: 2) {...}
```

Especificamos el número de bits por componentes del color. El anterior *media querie* se aplica para los dispositivos con 2 o más bits como componentes de color, acepta los prefijos min-/max-.

## 9 - color-index

```
@media all and (min-color-index: 1) {...}
```

Con esto especificamos el número de entradas en la tabla de conversión de color del dispositivo de salida, si el dispositivo no utiliza una tabla de conversión de color, su valor será cero, acepta los prefijos min-/max-.

## 10 - monochrome

```
@media all and (min-monochrome: 2) {...}
```

Básicamente con esta función especificamos que el dispositivo está en blanco y negro, y lo especificamos por el número de bits por píxel, acepta los prefijos min-/max-.

```
@media all and (monochrome) {...}
@media all and (min-monochrome: 1) {...}
```

En el anterior código especificamos 2 formas de aplicar los css especiales por *media queries* a todos los dispositivos monocromáticos es decir a blanco y negro.

## 11 - resolution

```
@media print and (min-resolution: 300dpi) {...}
```

Especificamos la densidad de píxeles de un dispositivo de salida, acepta los prefijos min-/max-.

Por ejemplo en el código anterior especificamos los css para los dispositivos con una resolución mínima de 300 ppp “puntos por pulgadas”.

## 12 - scan

```
@media tv and (scan: progressive) {...}  
/* o */  
@media tv and (scan: interlace) {...}
```

Especificamos los css para los procesos de escaneados utilizados por los dispositivos de salida de televisión, los valores pueden ser ***progressive*** o ***interlace***.

En resumen.

En CSS 2.1 utilizamos los *media types screen* y *print* para aplicar estilos distintos a la web; ahora en CSS3 tenemos la opción de *media queries*, que es mucho más abundante en funcionalidades, ya que nos permite crear estilos css alternos para los distintos dispositivos que pueden conectarse a internet y visualizar una web, ya no hay motivos para no hacer que nuestra web se vea bien en todas las pantallas.

Para más información sobre este tema: <http://www.w3.org/TR/css3-mediaqueries/>

Para ver los tamaños de los dispositivos móviles: <http://screensiz.es>



## TRANSFORM

La propiedad **transform** permite añadir nuevas formas de cambiar un elemento en 2D o 3D:

- **matrix(n,n,n,n,n,n)**: Define una transformación 2D, utilizando una matriz de seis valor.
- **matrix3d(n,n,n,n,n,n,n,n,n,n,n,n)**: Define una transformación 3D, utilizando una matriz 4x4 de dieciséis valores.
- **translate(x,y)**: Define una transformación 2D de traslación.



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE).

- **translate3d(x,y,z)**: Define una transformación 3D de traslación.
- **translateX(x)**: Define una traslación sólo en el eje X.
- **translateY(y)**: Define una traslación sólo en el eje Y.
- **translateZ(z)**: Define una traslación sólo en el eje Z.
- **scale(x,y)**: Define una transformación 2D de escala.
- **scale3d(x,y,z)**: Define una transformación 3D de escala.
- **scaleX(x)**: Define una transformación de escalera sólo en el eje X.
- **scaleY(y)**: Define una transformación de escalera sólo en el eje Y.
- **scaleZ(z)**: Define una transformación de escalera sólo en el eje Z.
- **rotate(angle)**: Define una transformación 2D de rotación, cuyo ángulo se especifica en el parámetro.
- **rotate3d(x,y,z,angle)**: Define una transformación 3D de rotación.
- **rotateX(angle)**: Define una transformación 3D de rotación en el eje X.
- **rotateY(angle)**: Define una transformación 3D de rotación en el eje Y.
- **rotateZ(angle)**: Define una transformación 3D de rotación en el eje Z.
- **skew(x-angle,y-angle)**: Define una transformación 3D de oblicuidad en el eje X e Y.
- **skewY(angle)**: Define una transformación 2D de oblicuidad en el eje Y.
- **perspective(n)**: Define una transformación 3D de perspectiva.

```
div {
    transform: rotate(45deg);
}
```

Ya hay muchas herramientas online para poder generar el código necesario para hacer estas transformaciones como <http://www.css3maker.com/css3-transform.html>

## TRANSICIONES

Con CSS 3 podemos realizar animaciones para hacer la transición de un estado a otro de un elemento con la propiedad **transition**.

Propiedad	Descripción	Valores
<b>transition-delay</b>	Especifica cuándo comienza la transición	[<tempo> initial   inherit ]
<b>transition-duration</b>	Especifica cuánto dura la transición en milisegundos	[<tempo> initial   inherit ]
<b>transition-property</b>	Especifica el nombre de la propiedad CSS afectada por la transición	[ none   all   <propiedad>   initial   inherit ]
<b>transition-timing-function</b>	Especifica la curva de velocidad del efecto.	[ ease   linear   ease-in   ease-out   ease-in-out   cubic-bezier()   initial   inherit ]
<b>transition</b>	Especifica las cuatro propiedades de golpe	[<property> <duration> <timing-function> <delay>   initial   inherit ]

Para utilizar esta propiedad, debemos definir **transition** en el estilo principal con el resto de atributos por defecto.

Luego hacemos otro estilo por el mismo elemento, normalmente asociado a un selector, con un cambio de una propiedad del estilo principal. Es ese cambio quien tendrá la animación de transición.



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

```
div {
    width: 100px;
    height: 100px;
    background: red;
    transition: width 2s;
}

div:hover {
    width: 300px;
}
```

Ya hay muchas herramientas online para poder generar el código necesario para hacer estas transiciones como <http://www.css3maker.com/css3-transition.html>

## ANIMACIONES

Con CSS 3 podemos realizar animaciones con la propiedad **animation**.

Propiedad	Descripción	Valores
<b>animation-name</b>	Especifica el nombre de @keyframes que se aplica al objeto	[keyframename   none   initial   inherit ]
<b>animation-duration</b>	Duración en segundos o milisegundos de la animación	[ <Xs>    <Xms>   initial   inherit ]
<b>animation-timing-function</b>	Especifica la curva de velocidad del efecto	[ linear   <b>ease</b>   ease-in   ease-out   cubic-bezier(n,n,n,n)   initial   inherit ]
<b>animation-delay</b>	Especifica cuándo comienza la transición	[ <Xs>    <Xms>   initial   inherit ]
<b>animation-iteration-count</b>	Especifica cuántas veces se debe reproducir la animación	[ <número> infinite   initial   inherit ]
<b>animation-direction</b>	Especifica el sentido de la animación en las repeticiones alternativas	[normal   reverse   alternate   alternate-reverse   initial   inherit ]

Estos atributos los ponemos en los estilos del objeto que queremos animar, y después definimos con la regla **@keyframes** el atributo(s) y valor(s) a animar. Dentro de **@keyframes** podemos poner un límite con **from** y **to**, o con porcentajes: **0%, 50%, 100%**.

```
div {
    width: 100px;
    height: 100px;
    background: red;
    position: relative;
    animation-name: mymove;
    animation-duration: 5s;
}
@keyframes mymove {
    from {left: 0px;}
    to {left: 200px;}
}
```

Ya hay muchas herramientas online para poder generar el código necesario para hacer estas animaciones como <http://www.css3maker.com/css3-animation.html>

## MAQUETACIÓN

Para poder distribuir los diferentes bloques en pantalla disponemos de diferentes técnicas empleando la estructura HTML y los atributos CSS.

### FLOAT

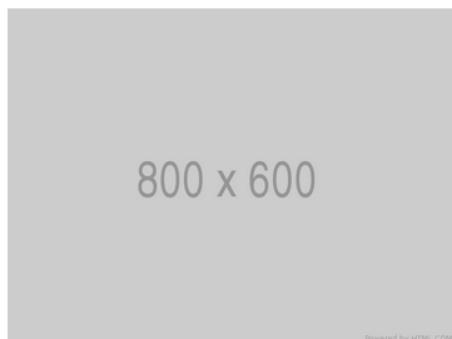
Una de las técnicas más clásicas es el uso de la propiedad **float**, donde indicamos a un bloque hacia donde se dispone respecto a su contenedor, y el contenido que viene a continuación se coloca a su lado.

Para que la propiedad **float** funcione correctamente, es necesario que el elemento que flote tenga también especificada la propiedad **width**.

Finalmente, para finalizar el atributo **float**, es necesario que otro elemento HTML al mismo nivel jerárquico, tenga el atributo **clear**.

Por ejemplo, el código HTML para mostrar que el bloque de imagen quede a la izquierda, el párrafo de texto flote a continuación a la derecha y la línea horizontal finalice el efecto de flotación:

### Título del artículo



La imagen destacada

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam nec erat ac ante fringilla molestie. Curabitur condimentum, nisi non pharetra maximus, nulla turpis interdum lorem, a gravida est quam vel est. In quis enim elit.

El código HTML sería éste:

```
<article>

    <h1>Título del artículo</h1>

    <figure>
        
        <figcaption>La imagen destacada</figcaption>
    </figure>

    <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam nec erat ac ante fringilla molestie. Curabitur condimentum, nisi non pharetra maximus, nulla turpis interdum lorem, a gravida est quam vel est. In quis enim elit.</p>

    <hr>

</article >
```

Y el código CSS sería éste:



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

```

body {
    font-family: Geneva, sans-serif;
}

figure {
    float: left;
    width: 33.33%;
    min-width: 300px;
    margin-top: 0;
}

figure img {
    max-width: 100%;
}

figcaption {
    color: #666;
    text-align: center;
}

hr {
    clear: left;
}

```

Si queremos que diferentes bloques floten uno al lado del otro, todos los bloques deben ser hermanos, es decir, tienen un mismo elemento contenedor, y todos deben tener la propiedad **float** e **width** especificada. Por último, es necesario que otro elemento hermano finalice la flotación con la propiedad **clear**, aunque podemos utilizar el pseudoelemento **::after** con la propiedad **content** para añadir contenido dinámicamente con CSS.

Por ejemplo, el código HTML para mostrar el contenido de una lista de definición en formato de tabla:

<b>Concepto 1:</b>	Definición 1
<b>Concepto 2:</b>	Definición 2
<b>Concepto 3:</b>	Definición 3
<b>Concepto 4:</b>	Definición 4

El código HTML sería éste:

```

<dl>

    <dt> Concepto 1: </dt>
    <dd> Definición 1 </dd>

    <dt> Concepto 2: </dt>
    <dd> Definición 2 </dd>

    <dt> Concepto 3: </dt>
    <dd> Definición 3 </dd>

    <dt> Concepto 4: </dt>
    <dd> Definición 4 </dd>

</dl >

```

Y el código CSS sería éste:



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

```

body {
    font-family: Geneva, sans-serif;
}

dl {
    margin: 1em;
    border: 1px solid #666;
    padding-top: 0.5rem;
}

dt, dd {
    float: left;
    box-sizing: border-box;
    padding: 0.5rem;
    border-bottom: 1px solid #666;
    margin: 0 0 0.5rem 0;
}

dt:last-of-type, dd:last-of-type {
    border-bottom: none;
    margin-bottom: 0;
}

dt {
    width: 35%;
    font-weight: bold;
}

dd {
    width: 65%;
}

dl::after {
    clear: left;
    display: block;
    content: "";
}

```

## FLEXBOX

Una técnica más moderna, pero ampliamente soportada por los navegadores es utilizar la propiedad **display** con el valor **flex**, para indicar al navegador que el contenido de un bloque se dispondrá de forma flexible. Con una serie de atributos adicionales, podemos controlar la colocación de los elementos interiores.

Propiedad	Descripción	Valores
<b>flex-direction</b>	Especifica la dirección de los elementos flexibles	[row   row-reverse   column   column-reverse   initial   inherit ]
<b>flex-wrap</b>	Especifica el desbordamiento de los elementos flexibles fuera de la caja flexible	[nowrap   wrap   wrap-reverse   initial   inherit ]
<b>flex-flow</b>	Especifica la dirección y el desbordamiento de los elementos flexibles	[<flex-direction>< flex-wrap>]
<b>align-items</b>	Especifica la colocación vertical de los elementos flexibles dentro de la caja flexible	[stretch   center   flex-start   flex-end   baseline   initial   inherit ]
<b>justify-content</b>	Especifica la colocación horizontal de los elementos flexibles dentro de la caja flexible	[flex-start   flex-end   center   space-between   space-around   space-evenly   initial   inherit ]
<b>flex</b>	Especifica la proporción y el tamaño mínimo de un elemento flexible dentro de la caja flexible	[ <proporción>   <longitud>   <porcentaje>   auto   initial   none   inherit]



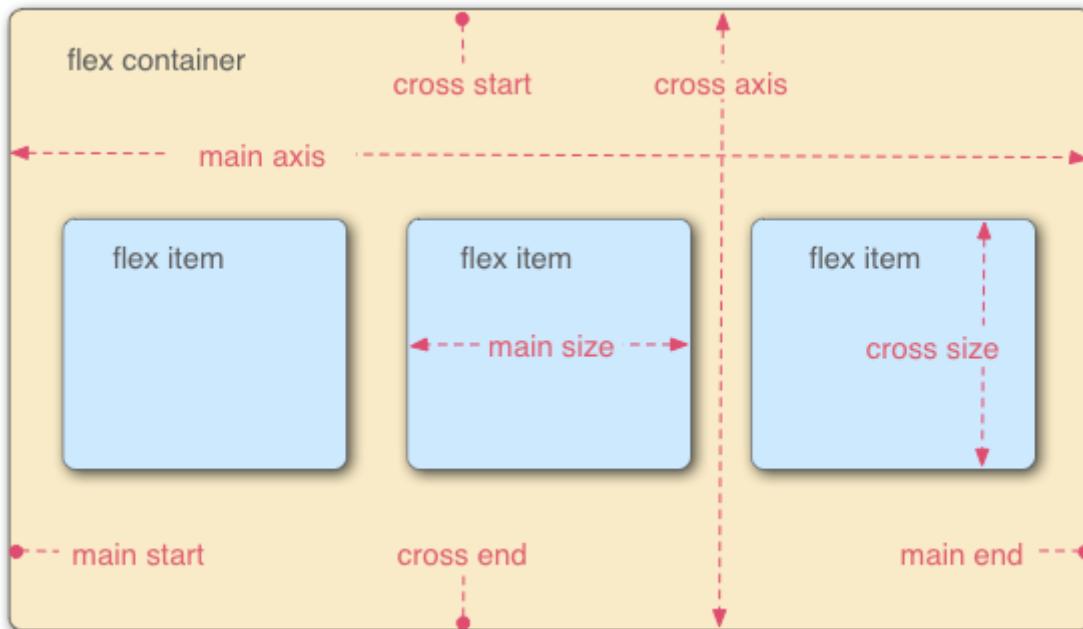
**Generalitat  
de Catalunya**



<b>align-self</b>	Especifica la colocación vertical de un elemento flexible dentro de la caja flexible	[stretch   center   flex-start   flex-end   baseline   initial   inherit ]
<b>order</b>	Especifica el orden de un elemento flexible dentro de la caja flexible	[number   initial   inherit ]

Para empezar debemos decidir qué elementos queremos configurar para que se comporten como cajas flexibles. Para ello, establecemos el valor **flex** por la propiedad **display** en el elemento padre de los elementos a los que afectará.

Los elementos que se disponen como cajas flexibles, se distribuyen a lo largo de dos ejes:



- El eje principal (*main axis*) es el eje que corre en la dirección en la que se disponen los elementos **flex** (p. ej., como filas a lo largo de la página o como columnas, hacia abajo de la página). El inicio y el final de este eje reciben los nombres de inicio principal (*main start*) y final principal (*main end*).
- El eje transversal (*cross axis*) es el eje que corre perpendicular a la dirección en la que se disponen los elementos flexibles. El inicio y el final de este eje se llaman inicio transversal (*cross-start*) y final transversal (*cross-end*).
- El elemento padre que está configurado con **display: flex** se llama **contenedor flex** (*flex container*).
- Los elementos que se disponen como cajas flexibles dentro del **contenedor flex** se llaman **elementos flexibles (flex items)**.

#### ¿Columnas o filas?

Flexbox proporciona una propiedad llamada **flex-direction**, que especifica en qué dirección discurre el eje principal (en qué dirección se disponen las cajas flexibles hijo); por defecto, está definida en **row**, que hace que los elementos flexibles se dispongan en una fila en la dirección del idioma predeterminado con el que funciona tu navegador (de izquierda a derecha, en el caso de un navegador en catalán).

Si se define como columna los elementos flexibles se disponen uno debajo del otro, como elementos de bloque.

### Ajuste

Un problema que surge cuando tienes un diseño de página con una anchura o altura fija es que los hijos del elemento flexbox pueden desbordar el contenedor y romper el diseño de la página. Por eso es necesario establecer la propiedad **flex-wrap: wrap**.

### Dimensión flexible de los elementos flexibles

La propiedad **flex** es un valor de proporción sin unidad que determina qué cantidad del espacio disponible a lo largo del eje principal ocupa cada elemento flexible. Si damos a los elementos flexibles un valor de 1 significa que todos ocuparán una cantidad igual del espacio que sobre después de que se hayan establecido elementos como el área de relleno (**padding**) y el margen (**margin**). Es una proporción, que significa que darle a cada elemento flexible un valor de 400.000 tendría exactamente el mismo efecto.

Además de la proporción, podemos añadir un tamaño exacto (en **px**, **%**, **em**) para establecer un mínimo de ancho.

### Alineación horizontal y vertical

También podemos utilizar propiedades flexbox para alinear elementos flexibles a lo largo de los ejes principales o transversales con **align-items** y **justify-content**.

La propiedad **align-items** dispone los elementos flexibles sobre el eje transversal.

- El valor predeterminado es **stretch**, que estira todos los elementos flexibles hasta ocupar todo el espacio del elemento padre en la dirección del eje transversal. Si el elemento padre no tiene un ancho fijo en la dirección del eje transversal, todos los elementos flexibles se harán tan largos como el elemento flexible más largo.
- El valor **center** mantiene las dimensiones intrínsecas de los elementos, pero los centrados a lo largo del eje transversal.
- También puedes tener valores como **flex-start** y **flex-end**, que alinean todos los elementos al principio y al final del eje transversal, respectivamente.

Puedes anular el comportamiento **align-items** para elementos flexibles individuales con la propiedad **align-self**.

La propiedad **justify-content** controla dónde se ubican los elementos flexibles sobre el eje principal.

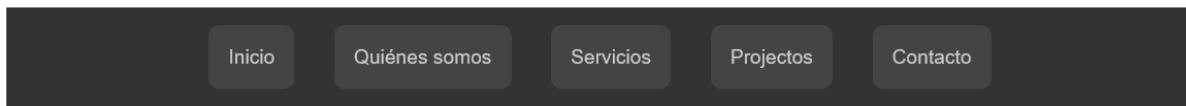
- El valor por defecto es **flex-start**, que coloca todos los elementos al principio del eje principal.
- Puedes utilizar **flex-end** para que se coloquen al final.
- El valor **center** coloca los elementos flexibles en el centro del eje principal.
- El valor **space-around** es útil porque distribuye todos los elementos sobre el eje principal de forma uniforme y deja un poco de espacio en cada extremo.
- Hay otro valor, **space-between**, que es muy similar a **space-around**, pero que no deja ningún espacio en los extremos.

### Ordena els elements flexibles

Flexbox también tiene la propiedad **order** que cambia el orden de disposición en pantalla de los elementos flexibles sin afectar al orden de origen en el código HTML. Esta es otra característica que es imposible hacer con los métodos tradicionales de diseño de páginas web.

- Por defecto, todos los elementos flexibles tienen un valor de orden (**order**) de **0**.
- Los elementos flexibles que están configurados con un valor de orden más alto aparecen más adelante en el orden de visualización que los elementos con valores de orden más bajo.
- Los elementos flexibles que tienen el mismo valor de orden aparecen en el orden de origen.
- Puedes establecer valores de orden negativos y hacer que los elementos aparezcan antes que los elementos con un orden **0**.

Probamos con un ejemplo de un menú de navegación:



```
<nav>
    <ul>
        <li><a href="#">Inicio</a></li>
        <li><a href="#">Quiénes somos</a></li>
        <li><a href="#">Servicios</a></li>
        <li><a href="#">Proyectos</a></li>
        <li><a href="#">Contacto</a></li>
    </ul>
</nav>
```

El código CSS sería éste:

```
nav {
    width:960px;
    margin:1em auto;
}

nav ul {
    margin:0;
    padding:0;
    height:80px;
    background:#333;
    list-style-type:none;
    display:flex; /* para indicar que la caja es flexible */
    flex-direction: row; /*para indicar la posición de los elementos flexibles */
    flex-wrap: wrap; /*para evitar el desbordamiento de los elementos flexibles*/
    align-items: center; /* para indicar la colocación vertical */
    justify-content: center; /* para indicar la colocación horizontal */
}

nav ul li {
    margin:0 1em;
}

nav a {
    color:#ccc;
    text-decoration:none;
```



**Generalitat  
de Catalunya**



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

```

        display:block;
        padding:1em;
        border-radius:1ex;
        background:#444;
    }

    nav a:hover {
        color:#fff;
        background:#666;
    }
}

```

## GRID

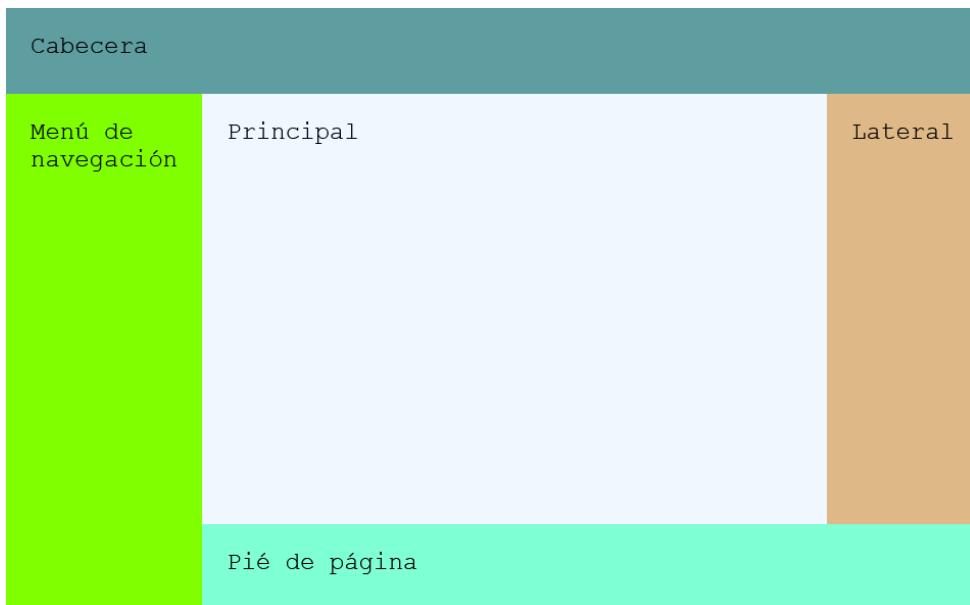
Otra técnica más moderna, pero sólo soportada por las últimas versiones de los navegadores, es utilizar la propiedad **display** con el valor **grid**, para indicar al navegador que el contenido de un bloque se dispondrá en una cuadrícula de filas y columnas. Por tanto, la principal diferencia respecto al **flexbox**, que sólo permite definir la colocación en el eje horizontal o vertical, es que el **grid** permite la colocación en los ejes horizontal y vertical. Con una serie de propiedades adicionales, podemos controlar la colocación de los elementos interiores.

Propiedad	Descripción	Valores
<b>grid-template-columns</b>	Especifica el número (y el ancho) de las columnas de la cuadrícula	<b>none auto max-content min-content length initial inherit</b>
<b>grid-template-rows</b>	Especifica el número (y altura) de las filas de la parrilla	<b>none auto max-content min-content length initial inherit</b>
<b>grid-template-areas</b>	Especifica las áreas de la cuadrícula, donde cada área es el nombre de los elementos hijos especificado con la propiedad <b>grid-area</b> .	<b>none itemnames</b>
<b>grid-template</b>	Resume las propiedades anteriores, distribuyendo las áreas en filas y columnas.	<b>none grid-template-rows / grid-template-columns grid-template-areas initial inherit</b>
<b>align-content</b>	Especifica la colocación vertical de los elementos dentro de la parrilla	<b>stretch   center   flex-start   flex-end   baseline   initial   inherit</b>
<b>justify-content</b>	Especifica la colocación horizontal de los elementos dentro de la parrilla	<b>flex-start   flex-end   center   space-between   space-around   space-evenly   initial   inherit</b>
<b>column-gap</b>	Especifica la distancia horizontal de los elementos dentro de la parrilla	<b>length normal initial inherit</b>
<b>row-gap</b>	Especifica la distancia vertical de los elementos dentro de la parrilla	<b>length normal initial inherit</b>
<b>gap</b>	Especifica la distancia de los elementos dentro de la parrilla	<b>row-gap column-gap initial inherit</b>

Probamos con un ejemplo de una estructura típica de una web:



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)



```
<body>

    <header>Cabecera</header>

    <nav>Menú de navegación</nav>

    <main>Principal</main>

    <aside>Lateral</aside>

    <footer>Pié de página</footer>

</body>
```

El código CSS sería éste:

```
body {
    font-family: 'Courier New', Courier, monospace;
    font-size: 22px;
    margin: 0;
}
header {
    grid-area: header;
    background-color: cadetblue;
    padding: 1em;
    box-sizing: border-box;
}
nav {
    grid-area: nav;
    background-color: chartreuse;
    padding: 1em;
    box-sizing: border-box;
}
main {
    grid-area: main;
    background-color: aliceblue;
    padding: 1em;
    box-sizing: border-box;
```



**Generalitat  
de Catalunya**



MINISTERIO  
DE EDUCACIÓN, FORMACIÓN PROFESIONAL  
Y DEPORTES



MINISTERIO  
DE TRABAJO  
Y ECONOMÍA SOCIAL



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

```

}
aside {
    grid-area: aside;
    background-color: burlywood;
    padding: 1em;
    box-sizing: border-box;
}
footer {
    grid-area: footer;
    background-color: aquamarine;
    padding: 1em;
    box-sizing: border-box;
}
@media (min-width:768px) {
    html {height: 100%;}
    body {
        height: 100%;
        overflow: hidden;
        display: grid;
        grid-template:
            "header header header" 1fr
            "nav main aside" 5fr
            "nav footer footer" 1fr /
            1fr 5fr 1fr;
    }
}

```

En este ejemplo, las diferentes áreas (*header*, *nav*, *main*, *aside* y *footer*) tienen la propiedad **grid-area** con un nombre especificado, siendo el *body* quien tiene la propiedad **display** con el valor **grid** y la propiedad **grid-template** con la estructura de 3x3.

En la primera fila del código ponemos las áreas que queremos que se vean en la primera fila de la pantalla, que en este caso sólo es la cabecera, así que lo ponemos tres veces para representar que va a ocupar los tres espacios. Y a continuación ponemos la altura que tendrá la primera fila: el valor **fr** representa una proporción respecto al resto de filas, que en este caso será 1/7 parte de la altura del *body*.

En la segunda fila del código ponemos las áreas que queremos que se vean en la segunda fila de la pantalla, que en este caso son el menú, el contenido principal y el lateral, así que los ponemos por orden para representar que ocuparán los tres espacios . Y a continuación ponemos la altura que tendrá la segunda fila: el valor **fr** representa una proporción respecto al resto de filas, que en este caso será 5/7 parte de la altura del *body*.

En la tercera fila del código ponemos las áreas que queremos que se vean en la tercera fila de la pantalla, que en este caso son el menú y el pie, así que los ponemos por orden para representar que el menú ocupará un espacio y el pie dos. Y a continuación ponemos la altura que tendrá la tercera fila: el valor **fr** representa una proporción respecto al resto de filas, que en este caso será 1/7 parte de la altura del *body*. La barra final indica que a continuación añadiremos las anchuras de las columnas.

En la cuarta fila del código ponemos las anchuras de las áreas: el valor **fr** representa una proporción respecto al resto de columnas, que en este caso será 1/7 parte de la anchura del *body*, 5/7 y finalmente 1/7.

## COLUMNAS

El diseño CSS de varias columnas permite una fácil definición de múltiples columnas de texto, al igual que en los periódicos:

  Lorem ipsum dolor sit amet,  
  consectetuer adipiscing elit, sed diam  
  nonummy nibh euismod tincidunt ut  
  laoreet dolore magna aliquam erat  
  voluptat. Ut wisi enim ad minim  
  veniam, quis nostrud exerci tation  
  ullamcorper suscipit lobortis nisl ut

  aliquip ex ea commodo consequat. Duis  
  autem vel eum iriure dolor in hendrerit  
  in vulputate velit esse molestie  
  consequat, vel illum dolore eu feugiat  
  nulla facilisis at vero eros et accumsan  
  et iusto odio dignissim qui blandit  
  praesent luptatum zzril delenit augue

  duis dolore te feugait nulla facilisi.  
  Nam liber tempor cum soluta nobis  
  eleifend option congue nihil imperdiet  
  doming id quod mazim placerat facer  
  possim assum.

```
.newspaper {  
  column-count: 3;  
  column-gap: 40px;  
  column-rule: 1px solid lightblue;  
}
```

Propiedad	Descripción	Valores
<b>column-count</b>	Especifica el número de columnas en las que se debe dividir un elemento	<b>auto</b>   <i>number</i>   <b>initial</b>   <b>inherit</b>
<b>column-fill</b>	Especifica cómo llenar columnas	<b>balance</b>   <b>auto</b>   <b>initial</b>   <b>inherit</b>
<b>column-gap</b>	Especifica el espacio entre las columnas	<b>normal</b>   <i>length</i>   <b>initial</b>   <b>inherit</b>
<b>column-rule</b>	Una propiedad abreviada para establecer todas las propiedades column-rule-*	column-rule-width column-rule-style column-rule-color   <b>initial</b>   <b>inherit</b>
<b>column-rule-color</b>	Especifica el color de la regla entre columnas	<b>color</b>   <b>initial</b>   <b>inherit</b>
<b>column-rule-style</b>	Especifica el estilo de la regla entre columnas	<b>none</b>   <b>hidden</b>   <b>dotted</b>   <b>dashed</b>   <b>solid</b>   <b>double</b>   <b>groove</b>   <b>ridge</b>   <b>inset</b>   <b>outset</b>   <b>initial</b>   <b>inherit</b>
<b>column-rule-width</b>	Especifica el ancho de la regla entre columnas	<b>medium</b>   <b>thin</b>   <b>thick</b>   <i>length</i>   <b>initial</b>   <b>inherit</b>
<b>column-span</b>	Especifica el número de columnas que debe abarcar un elemento	<b>none</b>   <b>all</b>   <b>initial</b>   <b>inherit</b>
<b>column-width</b>	Especifica un ancho óptimo sugerido para las columnas	<b>auto</b>   <i>length</i>   <b>initial</b>   <b>inherit</b>
<b>columns</b>	Una propiedad abreviada para establecer column-width y column-count	<b>auto</b>   <i>column-width column-count</i>   <b>initial</b>   <b>inherit</b>

Una forma práctica de aplicar las columnas es utilizar el atributo **columns** con los valores auto y anchura:

```
.newspaper {  
  column: auto 10rem;  
}
```

El navegador escoge automáticamente el número de columnas basándose en la anchura mínima especificada y el espacio disponible.



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

## VARIABLES

En CSS3, las variables (también conocidas como *Custom Properties*) son una herramienta muy útil para definir valores reutilizables que puedes utilizar a lo largo de tu hoja de estilos. Esto ayuda a mantener el código más organizado y fácil de mantener, especialmente en proyectos grandes. Las ventajas de las variables en CSS son claras:

- **Reutilización:** Puedes definir un color, fuente, o cualquier valor una vez y reutilizarlo múltiples veces.
- **Mantenimiento:** Si necesitas cambiar el valor de un color o un tamaño, solo tienes que hacerlo una vez, en el lugar donde se define la variable.
- **Personalización dinámica:** Las variables pueden cambiar en función del contexto (por ejemplo, en un tema oscuro o claro).

Las variables CSS son heredables, es decir, los valores se pueden pasar a los elementos hijos si no son redefinidos. Sólo funcionan con propiedades CSS (no se pueden utilizar en lugares como media queries o selectores directamente).

## FUNCIONAMIENTO DE LAS VARIABLES EN CSS3

1. Definir una variable: las variables se definen utilizando el prefijo `--`. Normalmente, se declaran dentro del selector `:root` para que sean accesibles globalmente a todo el documento. Por ejemplo:

```
:root {
  --color-principal: #3498db;
  --marge-general: 16px;
}
```

2. Utilizar una variable: para utilizar una variable, usas la función `var()` y pasas el nombre de la variable como argumento:

```
body {
  color: var(--color-principal);
  margin: var(--marge-general);
}
```

3. Variables con valores por defecto: puedes especificar un valor por defecto en caso de que la variable no esté definida:

```
h1 {
  color: var(--color-secondary, #e74c3c);
```

## MODO CLARO-OSCURO

Un ejemplo de uso de las variables puede ser aplicar colores diferentes a la página según estamos en un modo claro u oscuro definiendo los colores al `:root`.

La propiedad `color-scheme` indica con qué combinación de colores del sistema operativo se debe representar un elemento. Con el valor `light dark` el elemento se puede representar con el esquema de color claro u oscuro del sistema operativo (depende de la configuración del usuario).



Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

```
:root {  
    --color-clar: #efefef;  
    --color-fosc: #303030;  
    color-scheme: light dark;  
}  
  
/* en :root ya está definido el color-scheme: light dark; */  
body {  
    color:light-dark( var(--color-fosc), var(--color-clar) );  
    background-color: light-dark( var(--color-clar), var(--color-fosc) );  
}
```



**Generalitat  
de Catalunya**

Aquesta actuació està impulsada i subvencionada pel Servei Públic d'Ocupació de Catalunya (SOC) amb fons rebuts del Ministeri d'Educació, Formació Professional i Esport i del Servei Públic d'Ocupació Estatal (SEPE)

