

BLOQUE I - TEMA 3

Hojas de estilos: CSS (Cascade Style Sheet)

3.1 Introducción

CSS es un lenguaje de estilos pensado para controlar el aspecto de los documentos HTML. Es una forma de separar los contenidos y su presentación, es imprescindible para crear páginas web complejas.

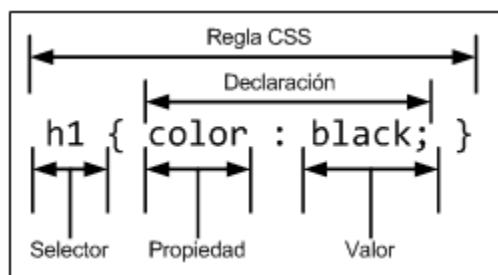
Al crear una página web, utilizamos HTML para definir los contenidos (designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.)

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

3.2 Funcionamiento básico

En CSS escribimos reglas:

```
body { background-color: #FF0000; }
```



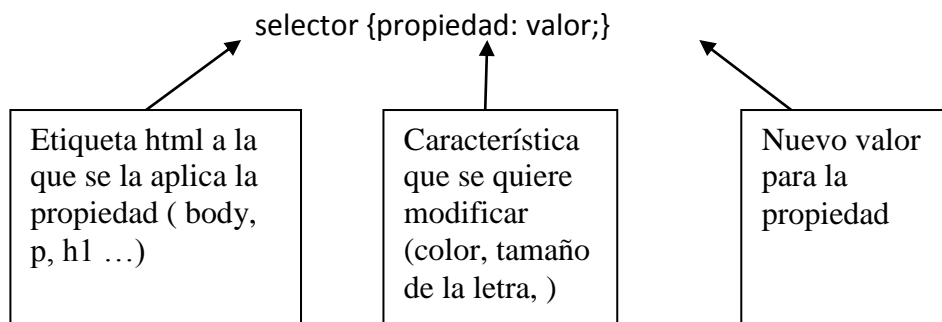
Javier Eguíluz Pérez

Regla: cada uno de los estilos que componen una hoja de estilos CSS.

Cada regla está compuesta de:

una parte de *"selectores"*
un símbolo de *"llave de apertura"*{
otra parte denominada *"declaración"*
y por último, un símbolo de *"llave de cierre"*}

- Selector: indica el elemento o elementos HTML a los que se aplica la regla CSS.
- Declaración: especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS.
- Propiedad: permite modificar el aspecto de una característica del elemento.
- Valor: indica el nuevo valor de la característica modificada en el elemento.



Existen tres opciones para incluir esas reglas:

1. Incluir css en las etiquetas HTML: se especifican las características en la etiqueta en la que queremos aplicarlas. Es el peor, porque si queremos cambiar un estilo tendremos que hacerlo en todas las etiquetas.
2. Incluir css en la cabecera de la página: Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta <style>de HTML y solamente se pueden incluir en la cabecera del documento
3. Definir CSS en un archivo externo: Un archivo de tipo CSS no es más que un archivo simple de texto cuya extensión es css .Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.

3.3. Aplicación directa en etiquetas

Para aplicar un estilo a una etiqueta concreta, usaremos la sintaxis:

```
<etiqueta style="propiedad1:valor;...;propiedadN:valor"> ... </etiqueta>
```

Ejemplo:

```
<p style="font-size:14pt;color:red"> Este texto es más grande y está en rojo</p>.
```

Esta forma es la menos usada para insertar estilos en un documento.

3.4 Redefinición de etiquetas

Si queremos definir estilos que se apliquen a todas las etiquetas del documento, es decir, queremos que todo el documento tenga un cierto tipo de letra, que las tablas tengan otro, que las cabeceras tengan un color determinado, para ello, definiremos estilos globales por medio de la etiqueta `<style> ...</style>` en la cabecera del documento como sigue:

```
<head>
  <title>Titulo de la pagina</title>
  <style type="text/css">
    Etiqueta1, Etiqueta2 {propiedad1:valor; }
    Etiqueta3 {
      propiedad1:valor;
      propiedad2 :valor;
    }
    Etiqueta4 {
      propiedad1 : valor;
      propiedad : valor;
    }
  </style>
</head>
```

Podemos aplicar el mismo estilo a varias etiquetas, escribiéndolas separadas por comas y, a continuación, la especificación del estilo según pares `propiedad : valor` separados por punto y coma y encerrados entre llaves `{}`.

Ejemplos:

```
<html>
  <head>
    <style type="text/css">
      p{
        color:red;
        text-align:center;
      }
    </style>
  </head>
  <body>
    <p>Hello World!</p>
    <p>El formato de este texto está hecho con CSS</p>
    Este texto no tiene formato
  </body>
</html>
```

Otro ejemplo:

```
<html>
  <head>
    <title> Ejemplo con bloque de estilo </title>
    <style type="text/css">
      body {
        font-family:Verdana,sans-serif;
        font-size:x-small;
        margin-left:0.25in;
        margin-right:0.25in
      }
      h2 {
        font-family:Arial;
        font-size:14pt;
        color:red;
      }
      p, td {
        color:olive;
      }
    </style>
  </head>
  <body>
    <p>Estoes un párrafo</p>
    <h2>Estoes un título</h2>
    <table>
      <tr><td>Celda (1,1)</td><td>Celda (1,2)</td></tr>
    </table>
  </body>
</html>
```

3.5 Separar HTML de CSS

Podemos definir los estilos en un fichero distinto al documento HTML, y después llamarlo desde el propio documento HTML. Esto lo haremos con la siguiente etiqueta, dentro de la cabecera del documento (entre `<head>` ...`</head>`):

```
<link rel="stylesheet" type="text/css" href="URL_Hoja.css">
```

```
<html>
  <head>
    <title>Ejemplo con hoja de estiloexterna</title>
    <link rel="stylesheet" type="text/css" href="ejemplo.css">
  </head>
  <body>
    <p>Estoes un párrafo</p>
    <h2>Estoes un título</h2>
    <table>
      <tr><th>Cabecera 1</th><th>Cabecera 2</th></tr>
      <tr><td>Celda (1,1)</td><td>Celda (1,2)</td></tr>
    </table>
  </body>
</html>
```

Si quisiéramos que otros documentos tuvieran el mismo estilo, sólo tendríamos que usar la etiqueta link para aplicarlos. Esa es la ventaja de las hojas de estilo externas.

En las hojas css se pueden introducir comentarios usando `/* */`.

3.6 SELECTORES

Un selector nos indica a quien le aplicamos el estilo. Una misma regla se puede aplicar sobre varios selectores y un mismo selector se puede utilizar en varias reglas.

Selector universal

Se utiliza para seleccionar todos los elementos de una página.

El siguiente ejemplo elimina el margen y el relleno de todos los elementos:

Ejemplo:

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

Selector de etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector.

Ejemplos

```
p {color: green;}
```

Si se quieren aplicar los mismos estilos a varias etiquetas diferentes, los selectores se separan por comas:

```
h1,h2,h3 {  
    color:blue;  
    font-family: Verdana;  
}
```

Si a continuación escribimos la siguiente regla:

```
h1 { font-size: 2cm;}
```

Estamos diciendo que los títulos h1, h2 y h3 se muestren en azul, con fuente Verdana y los h1 además tienen un tamaño de 2cm.

```
span {color:red}
```

```
<p>Te voy a explicar una <span>etiqueta html nueva</span>para ti. Hasta ahora no  
la hemos utilizado</p>
```

```
...
```

```
<h1>Este es otro sitio donde la coloco<span>Aquí también</span></h1>
```

Si en lugar de coma ponemos un espacio en blanco nos estaremos refiriendo a los elementos contenidos en otros:

```
h1 span{color: red}
```

Selector de clase

Puede suceder que la definición de un estilo concreto queramos aplicarla a otras etiquetas. Con este fin se definen las clases.

Una clase es una definición de un estilo que en principio no está asociado a alguna etiqueta HTML, pero que podemos asociar, en el documento, a etiquetas concretas.

Para ello, en primer lugar definimos la clase (en el bloque de estilos) como un estilo más, de esta forma:

```
.Nombre_de_la_Clase {propiedad1:valor;...;propiedadN:valor}
```

Ahora, para aplicar el estilo de una clase a una etiqueta concreta, utilizaremos el parámetro class como sigue:

```
<etiqueta class="Nombre_de_la_Clase"> ...</etiqueta>
```

dondeNombre_de_la_Clase es el nombre que le hemos dado a la clase, sin el punto.

Ejemplo:

```
<html>
  <head>
    <title> Ejemplo de uso de clases </title>
    <styletype="text/css">
      p {
        color:red;
        font-family:Verdana,sans-serif;
        font-size:x-small
      }
      .Azul {color:blue}
    </style>
  </head>
  <body>
    En este ejemplo vamos a ver cómo se aplican las clases.
    <h4>Este es un título normal</h4>
    <h4 class="Azul">Este es un título con clase</h4>
    <p>Este es un párrafo normal</p>
    <p class="Azul">Este es una párrafo con clase</p>
  </body>
</html>
```

Selector de id

Cualquier etiqueta HTML puede tener como parámetro la etiqueta id seguida de un nombre, por ejemplo:

```
<etiqueta ID="Nombre"> ...</etiqueta>
```

Este "Nombre" debe ser único en el documento HTML (es decir, no debe haber dos etiquetas con el mismo id), puesto que nos servirá para tratarla (si lo necesitamos) desde JavaScript.

Para definir un estilo mediante un id, escribimos # seguido del nombre del id (en un bloque de estilo o en la hoja externa):

```
#Nombre_del_ID {propiedad1:valor;...;propiedadN:valor}
```

Pseudoclases

Las pseudoclases de que disponemos son:

a: link Nos dice el estilo de un enlace que no ha sido visitado.

a:visited Nos dice el estilo de un enlace que ha sido visitado.

a:active Nos dice el estilo de un enlace que está siendo pulsado.

a:hover Nos dice el estilo de un enlace sobre el que está pasando el ratón.

Por ejemplo, si deseásemos que apareciesen todos los enlaces sin subrayar, definiríamos los siguientes estilos:

```
a:link,a:visited,a:active {text-decoration:none; }
```

3.7 PROPIEDADES

Unidades de medida

Las medidas en CSS se emplean para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto.

Todas las medidas se indican como un valor numérico entero o decimal seguido de una unidad de medida (sin ningún espacio en blanco entre el número y la unidad de medida). Si no ponemos unidad de medida se ignora dicha propiedad.

CSS divide todas las unidades de medida en :

Absolutas: Su valor es el indicado.

- **pt** (puntos), 0.35 mm
- **in** (pulgadas), 2.54 cms
- **mm** (milímetros),
- **cm** (centímetros).

Relativas: Su valor depende de otra medida.

- **em** (el alto de la M mayúscula), relativa respecto a la letra empleada (1em significa igual tamaño)

- **px** (píxel), relativa respecto a la pantalla del usuario.

Porcentajes: Son un valor numérico más el símbolo del porcentaje %. Se suele usar para definir el alto y ancho y representan el porcentaje de la medida de su elemento padre.

Normalmente se utilizan píxel y porcentajes para definir el tamaño de los elementos de las páginas y em para el tamaño de letra de los textos.

Colores

Se pueden indicar de varias maneras, igual que en HTML.

Fuentes

Estas propiedades se aplican al texto en cualquier elemento

- **color** : color de la letra
- **font-family**: tipo de letra
- **font-size**: tamaño
 - número y unidad de medida
 - alguno de los siguientes valores: xx-small, x-small, small, medium, large, x-large, xx-large,
- **font-weight**: letra negrita
 - bold
 - número (por ejemplo 900)
- **font-style** : letra cursiva
 - italic, oblique (se usan indistintamente)

Ejemplo

```
p{
    color: white;
    font-family: Verdana, arial;
    font-size: x-large;
    font-style: italic;
}
```

Sólo podremos utilizar aquellas fuentes que tengamos instaladas y nuestro navegador reconozca. Si no lo hace, utilizará la que tenga por defecto y no la que le indicamos. Es muy recomendable, poner varias familias de fuentes en la propiedad font-family y terminar la lista con una genérica(todos los navegadores las reconocen)

Valores de las familias genéricas son:

- 'serif' (Ej.: Times)
- 'sans-serif' (Ej.: Helvetica, Arial)
- 'cursive' (Ej.: Zapf-Chancery, Comic sans)
- 'fantasy' (Ej.: Western, Impact)
- 'monospace' (Ej.: Courier)

Texto

- **Text-align:** alineación de texto e imágenes contenidos en h1,h2...p, div , table (td)
 - Left, center, right, justify
- **Text-decoration:** subrayado del texto
 - underline: subrayado
 - overline: linea por encima
 - line-through: tachado
- **Text-transform:**
 - capitalize: primera en mayúsculas
 - lowercase: minúsculas
 - uppercase: mayúsculas
- **Vertical-align :**alineación vertical. Sólo lo utilizaremos en etiqueta td
 - top
 - middle
 - bottom
- **Letter-spacing:** separa las letras unas de otras. Su valor es un número. Cuanto más alto es dicho número, mayor es la separación entre las letras.
- **Text-shadow:** le aplica sombra al texto. Es necesario darle la posición horizontal y la vertical. También puede darse el color

Ejemplo: text-shadow: 2px 2px red

Fondos

CSS define propiedades para establecer el fondo de cada elemento

- `background-color`: Color de fondo
- `background-image`: Imagen de fondo

`background-image: url("nombre del archivoimagen")`

- `background-repeat`: Repetición de la imagen de fondo. Sus posibles valores
 - `repeat`
 - `repeat-x`
 - `repeat-y`
 - `no-repeat`
- `background-attachment` y `background-size`: la imagen ocupará toda la pantalla
 - `background-attachment: fixed;`
 - `background-size: 100% 100%;`

Si utilizamos la propiedad `background-size: cover` no necesitamos `background-attachment: fixed`; pero la imagen cubre toda la pantalla y podría no verse entera.

Estas propiedades se pueden aplicar a `body`, `table`, `td`, `div`

Consejo: siempre que coloquemos una imagen de fondo, debemos poner también un color de fondo cercano al color de la imagen.

Esto se debe a que, al colocar una imagen de fondo, el texto de la página debemos colocarlo en un color que contraste suficientemente con dicho fondo. Si la imagen no se carga, por cualquier cuestión (Por ejemplo tener deshabilitada la carga de imágenes) puede que el texto no contraste lo suficiente con el color de fondo por defecto de la web.

Por ejemplo: Si la imagen de fondo es oscura, tendremos que poner un texto claro para que se pueda leer. Si el visitante que accede a la página no ve la imagen de fondo, le saldrá el fondo por defecto, que generalmente es blanco, de modo que al tener un texto con color claro sobre un fondo blanco, nos pasará que no podremos leer el texto

Ejemplo

```
body{
    background-color:#ff2233;
    background-image: url("fondo3.jpg");
    background-repeat:no-repeat;
    background-attachment: fixed;
    background-size: 100% 100%;/*background-size:cover;*/
}
```

Márgenes

- `margin-top`: margen superior
- `margin-right`: margen derecho
- `margin-bottom`: margen inferior
- `margin-left`: margen izquierdo

Tamaño

- `width`: ancho (medimos en px o %)
- `height`: alto (medimos en px o %)

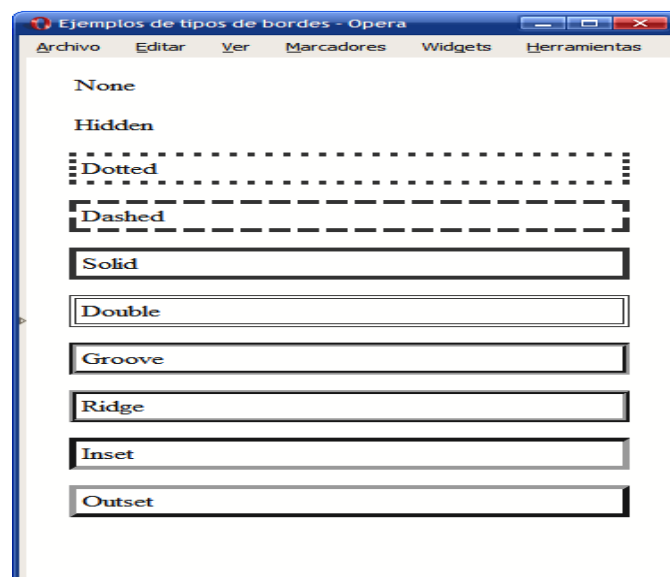
Listas

- `list-style-type`: tipo de viñeta. Los posibles valores
 - `disc` ●, `circle` ○, `square` ■, `decimal` (1,2,3..)
 - `decimal-leading-zero` (01,02..)
 - `lower-latin` (a,b,c ..),
 - `Upper-latin` (A,B,C...),
 - `lower-roman` (i,ii,iii..),
 - `upper-roman` (I,II,III..)
- `list-style-image`: en lugar de viñetas, ponemos una imagen.

`list-style-image: url("imagen");`

Tablas

- `border-style`: estilo del borde. Puede tomar uno de los sgtes valores



Todos los bordes aparecerán con el estilo indicado, si queremos que sólo lo haga alguno de ellos indicaremos cual

```
border-top-style: double;
```

- **border-color:** color del borde. Si queremos un borde concreto, lo indicamos

```
border-top-color:red;  
border-bottom-color:blue;
```

- **border-width:** ancho del borde. Su valor es una medida

```
border-width: 2px;
```

todos los bordes tendrán el mismo ancho, si se quiere que cada uno de ellos sea diferente se aplica el estilo a ese concreto

```
border-top-width:3px;
```

Se pueden poner estas tres propiedades juntas y se aplican a la vez

```
border: grosor color estilo;
```

Si queremos poner bordes exteriores le aplicamos el borde a la tabla

Si queremos poner bordes interiores le aplicamos el borde a las celdas

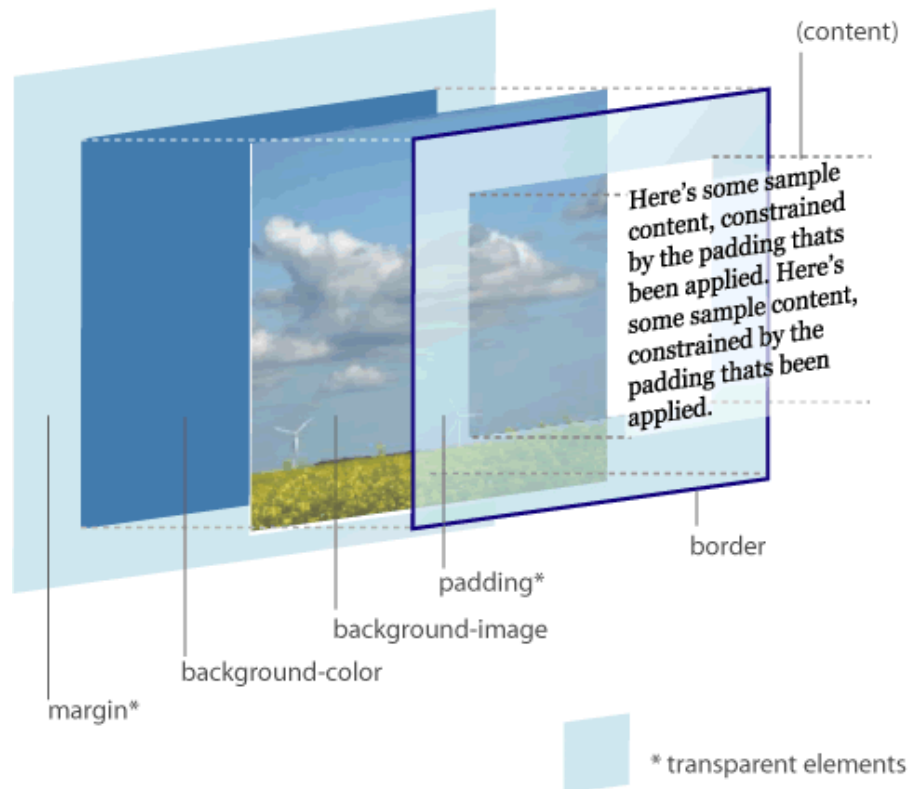
Si aplicamos ambos bordes pero no queremos que se vean dobles utilizamos la propiedad border-collapse

```
border-collapse: collapse;
```

Cajas

Todos los elementos html se representan mediante cajas rectangulares. Lo elementos de estas cajas son:

THE CSS BOX MODEL HIERARCHY



- **width**: ancho (medimos en px o %)
- **height**: alto (medimos en px o %)
- **margin-top**: margen superior
- **margin-right**: margen derecho
- **margin-bottom**: margen inferior
- **margin-left**: margen izquierdo
- **Padding-top** : relleno superior
- **Padding-right**: relleno derecho
- **Padding-bottom**: relleno inferior
- **Padding-left**: relleno izquierdo

- **border-style**: igual que en las tablas
- **border-radius**: redondea los bordes de una caja

`border-radius: 3px;`

Redondea todos, si quiero sólo alguno concreto tengo que indicarlo

`border-top-left-radius` ,`border-top-right-radius`, `border-bottom-left-radius` ,
`border-bottom-right-radius`

- **box-shadow**: sombra

`box-shadow: 10px 10px 50px red;`

3.8 EJEMPLOS ÚTILES

Cambiar el color de una fila de una tabla cuando pasamos el ratón

```
table tr: hover{  
    background: #FFFF66;  
}
```

Centrar el contenido de la página independientemente de la pantalla utilizada

Es una práctica habitual en el diseño de páginas elegir un color o imagen de fondo y centrar todo el contenido de la misma en la pantalla. Lo que realmente se hace es meter todo el html de la página dentro de un div y aplicarle css al mismo. Se establecen sus márgenes derecho e izquierdo a auto. De esta manera no se centra como tal, sino que se le obliga a que deje el mismo espacio a ambos lados. Para que esto tenga efecto, es obligatorio dar un valor a la propiedad width de dicho div

Ejemplo:

```
<html>  
  <head>  
    <title> Ejemplo con bloque de estilo </title>  
    <style type="text/css">  
      body{  
        background-color:black;  
      }  
      div{  
        margin-top:50px;
```

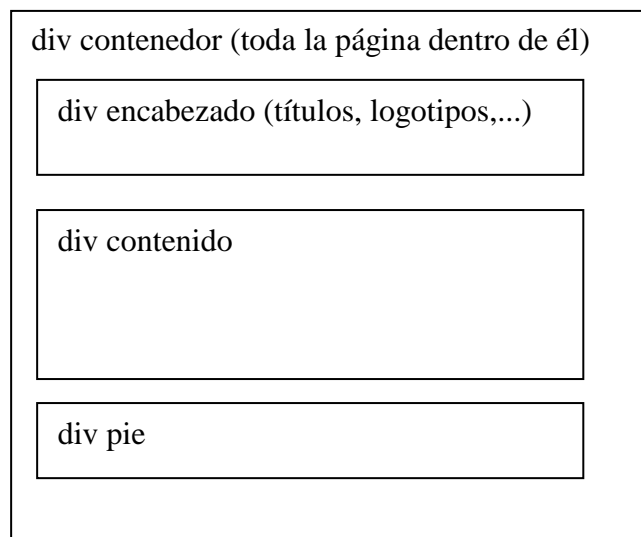
```
        width:70%;
        margin-left:auto;
        margin-right:auto;
        height:80%;
        background-color: #708090;
    }

</style>
</head>
<body>
<div>

</div>

</body>
</html>
```

Se suelen dividir las páginas a su vez en un div para el encabezado (títulos,...), un div para el contenido y otro para el pie (firma,...)



Menús

Los menús suelen hacerse a partir de listas (ul). Los elementos (li) estarán compuestos de enlaces. Se puede utilizar la propiedad display: inline para que los elementos queden en una línea.

```
<head>
  <style type="text/css">
    #menu a{
      height: 20px;
      font-family: Helvetica;
      font-size: 18px;
      color: #CCCCCC;
    }

    #menu ul {
      margin: 0;
      padding: 0;
    }

    #menu ul li {
      display: inline;
    }
  </style>
</head>

<body>

  <h1>Ejemplo de Menú de Navegación con HTML y CSS</h1>

  <div id="menu">
    <ul>
      <li><a href="#">Sección 1</a></li>
      <li><a href="#">Sección 2</a></li>
      <li><a href="#">Sección 3</a></li>
      <li><a href="#">Sección 4</a></li>
    </ul>
  </div>

</body>
</html>
```

Otro menú

```
<html>
<head>
<title>Ejemplo</title>
  <style>
```

```
#izquierda {
width: 170px;
float: left;
}
#menu {
padding:0;
margin:0 auto;
list-style-type:none;
width:160px;
height:312px;
}
#menu li a {
height:30px;
width:160px;
float: left;
border-bottom:2px solid #fff;
overflow: hidden;
}
#menu li a img {
border:0;
}
#menu li a:hover {
height:120px;
}
</style>

</head>
<body>
<div id="izquierda">
<ul id="menu">
<li><a href="#"><imgsrc="img/1.jpg" alt="#" title="#" /></a></li>
<li><a href="#"><imgsrc="img/2.jpg" alt="#" title="#" /></a></li>
<li><a href="#"><imgsrc="img/3.jpg" alt="#" title="#" /></a></li>
<li><a href="#"><imgsrc="img/4.jpg" alt="#" title="#" /></a></li>
<li><a href="#"><imgsrc="img/5.jpg" alt="#" title="#" /></a></li>
<li><a href="#"><imgsrc="img/6.jpg" alt="#" title="#" /></a></li>
<li><a href="#"><imgsrc="img/7.jpg" alt="#" title="#" /></a></li>
</ul>
</div>

</body>
</html>
```

Contenido multimedia

Vídeo

HTML5 permite insertar contenido multimedia, audio y vídeo.

```
<video src="video.mp4"> Tu navegador no  
soporta el elemento vídeo </video>
```

Atributos

poster: permite insertar una imagen de sustitución cuando el vídeo no está disponible.

controls: se incluyen controles de play, pause...

width y height: ancho y alto, se ajusta al valor y el otro se calcula automáticamente manteniendo la proporción.

loop: reproduce el vídeo continuamente .

autoplay: reproduce el vídeo automáticamente.

preload: carga el vídeo

```
<video src="video.mp4" poster="videojpg"  
autoplay loop controls></video>
```

e/s/d/
madrid

Es posible incluir diferentes formatos de un archivo para que el vídeo pueda ser soportado en distintos navegadores, se emplea la etiqueta **<source>**

```
<video width="640" height="360" controls preload>  
<source src="video.mp4" type='video/mp4; codecs="avc1,mp4a"' />  
<source src="video.ogv" type='video/ogg;  
codecs="theora,vorbis"' />  
<source src="video.webm" type='video/webm; codecs="vp8,vorbis"' />  
</video>
```

Formatos y compatibilidad de los navegadores:

https://developer.mozilla.org/es/docs/Web/HTML/Formatos_admitidos_de_audio_y_video_en_html5

Cada navegador, cada dispositivo, soporta diferentes formatos de vídeo. La solución es convertir el vídeo a los diferentes formatos de vídeo, que puede ser un proceso complejo y que requiere tiempo.

Una solución puede ser emplear **YouTube** para reproducir vídeo.

Sube el vídeo a YouTube, toma nota de su identificador o copia el código de inserción que facilita YouTube.

```
<iframe width="640" height="360"  
src="https://www.youtube.com/embed/CFp_8Sah6YU" frameborder="0" allowfullscreen>  
</iframe>
```

Los parámetros de YouTube:

autopause , valor **0** para tener los controles visibles y **1** si prefieres que se oculten cuando el vídeo se está reproduciendo.

autoplay, **0** si quieres que el vídeo se reproduzca automáticamente, **1** si espera al play.

controls, con **0** los controles no se muestran con **1** si y en ambos el vídeo se descarga el vídeo, con **2** los controles se muestran pero el vídeo no se descarga hasta hacer play.

Loop, **0** sólo una vez y **1** para reproducción infinita.

e/s/d/

Audio

El audio EN html4 se inserta de manera similar:

```
<audio src="audio.mp3" controls autoplay loop >
</audio>
```

El formato mp3, no es un formato abierto, para maximizar la compatibilidad usaremos varios formatos con **<source>**

```
<audio controls>
  <source src="audio.ogg" type="audio/ogg">
  <source src="audio.mp3" type="audio/mpeg">
  <source src="audio.wav" type="audio/wav">
</audio>
```

e/s/d/
madrid

3.9 FORMULARIOS

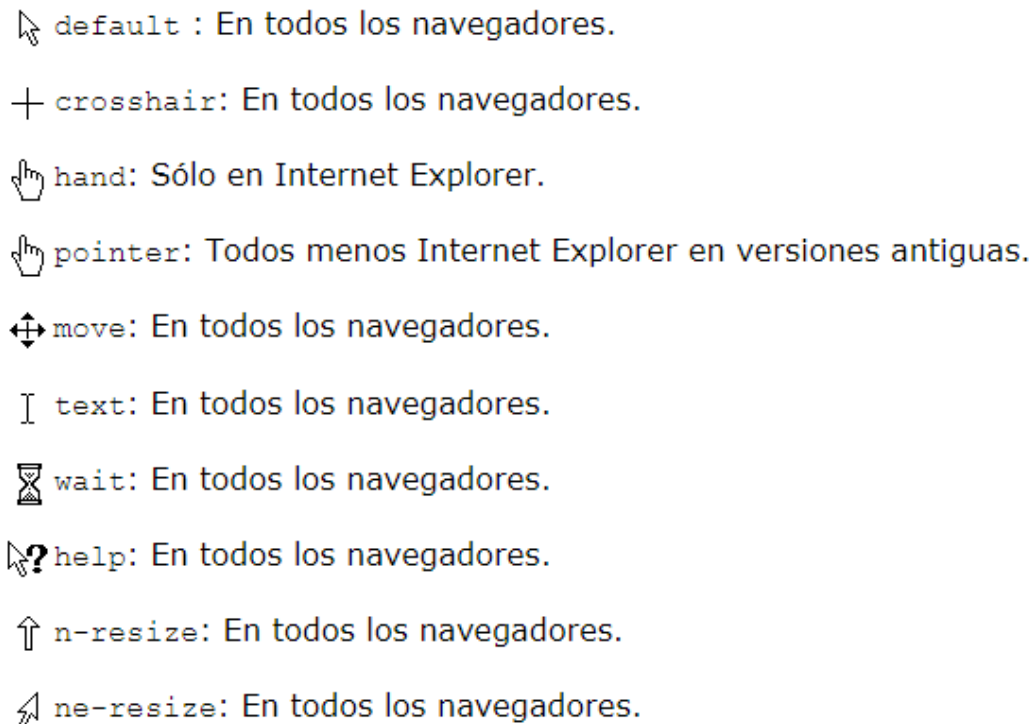
Hoy en día, los formularios son elementos muy frecuentes en las aplicaciones web, podemos mejorar su aspecto usando todas las propiedades que hemos visto con anterioridad, veamos algunas nuevas:











box-sizing: es una propiedad CSS para cambiar el modelo de caja por defecto de los navegadores. El ancho de un elemento cambia si se le aplica un borde o un padding. Cuando especificas ancho y alto de una caja, no incluyes en esas propiedades el borde o el padding, esto supone que el tamaño de la caja realmente cambiará cuando apliques todas las propiedades porque sumará al ancho tanto el padding como el borde.

Así funcionan los navegadores, el valor de esta propiedad es content-box, pero podemos cambiar el comportamiento eligiendo para esta propiedad su valor border-box. De esta manera el ancho total es el que especificamos en la propiedad y a partir de ahí calcula bordes y padding

cursor: podemos cambiar el aspecto del cursor cuando el ratón pasa por encima del elemento al que le aplicamos esta propiedad

Podemos usar los predefinidos

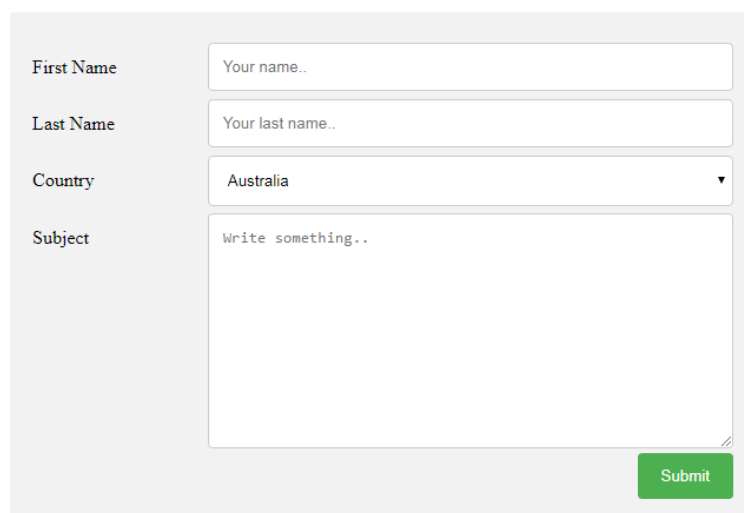


| | |
|---|---|
|  | <code>default</code> : En todos los navegadores. |
|  | <code>crosshair</code> : En todos los navegadores. |
|  | <code>hand</code> : Sólo en Internet Explorer. |
|  | <code>pointer</code> : Todos menos Internet Explorer en versiones antiguas. |
|  | <code>move</code> : En todos los navegadores. |
|  | <code>text</code> : En todos los navegadores. |
|  | <code>wait</code> : En todos los navegadores. |
|  | <code>help</code> : En todos los navegadores. |
|  | <code>n-resize</code> : En todos los navegadores. |
|  | <code>ne-resize</code> : En todos los navegadores. |

Estos son algunos ejemplos, hay más

o utilizar una imagen nuestra `URL("ruta_imagen")`

Hoy en día, el diseño más utilizado es bootstrap, veamos un ejemplo de un form que es responsive



First Name

Last Name

Country

Subject

Véase código HTML Y CSS en w3schools.com