

Resumen CSS/HTML

Josep Garcia Garcia

Índice de contenido

Antes de comenzar.....	3
Reglas básicas en CSS.....	3
Agrupación de Selectores y Declaraciones.....	4
Alineación.....	4
Transformación del texto.....	5
Decoración del texto.....	6
[COLORES] Clasificación de los elementos y Colores.....	7
Estilo en los enlaces.....	8
Comentarios en HTML y CSS.....	9
Estilos en los párrafos <p>.....	9
Estilos en los encabezados <H1> <H2> <H3>.....	9
La etiqueta <embed>.....	9
Elementos para enfatizar 	10
Selector universal * (asterisco).....	10
Estilos en las listas.....	11
La etiqueta <LINK>.....	12
La etiqueta <STYLE> y la directiva @import.....	13
Identificación y agrupación de elementos <CLASS> <ID>.....	14
La etiqueta	16
La importancia de !import en css.....	16
¿Qué son las capas? La etiqueta <DIV>.....	18
La propiedad display.....	18
La propiedad position.....	19
La propiedad float.....	21
La propiedad clear.....	23
Capa sobre capa con la propiedad z-index.....	23
Ejemplos con capas.....	24
Creación de una página web.....	28
¿ Qué son las palabras claves (Keywords) ?.....	29
Nombre de dominio.....	29
Alojamiento (hosting).....	30
Etiquetas META.....	31
Mapa visual del sitio (para usuarios).....	31
Mapa XML del sitio (para buscadores).....	32
Robots.txt.....	32
User-agent: Todos los bots de los buscadores.....	32
Allow: Permitir que se rastree todo el sitio.....	32
Disallow: No examinar el directorio_privado (contiene información sensible).....	32
Sitemap: URL del sitemap.xml.....	32
Usabilidad y Accesibilidad.....	33
Un poco de SEO (Search Engine optimization).....	34
Optimización de la web para buscadores.....	35
A tener en cuenta.....	35
Fuentes.....	36
Otros recursos.....	37
Resumen CSS.....	38

Antes de comenzar....

En la web hay miles de manuales CSS y HTML, para redactar este documento me he valido de muchos de ellos, copiando muchas de sus partes y algunas de sus imágenes ;). Pero no todo ha sido copiar, también he redactado algún que otro párrafo...

Os animo a que visitéis cada uno de los enlaces de la sección **FUENTES** (donde están los enlaces a todos los manuales que he utilizado para el resumen) si queréis ampliar un poco más la información de alguno de los temas que se tratan.

Reglas básicas en CSS

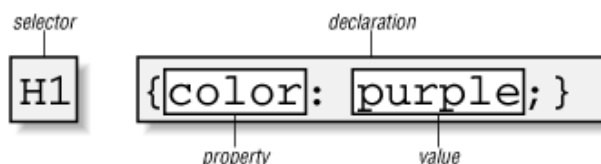
Con CSS podemos aplicar reglas de estilo a un tipo determinado de elementos dentro del documento HTML, así podemos conseguir el mismo efecto que utilizando el marcado tradicional:

```
<h2><font color="red">Texto de prueba</font></h2>
```

Pero con el mínimo esfuerzo:

```
H2 {color: red;}
```

Estructura de una regla



Si la definición del valor de una propiedad permite el uso de más de una palabra, estas estarán separadas por espacios en blanco. Además existen unos pocos ejemplos en los que se pueden utilizar otros tipos de elementos dentro de la declaración. Es el caso de la propiedad FONT:

```
H2 {font: large/150% sans-serif;}
```

Esta declaración permite definir además del tamaño de la fuente, el grosor de la línea en la que está el texto.

Agrupación de Selectores y Declaraciones

Normalmente, se da el caso en que dos elementos del documento HTML comparten la definición de la misma propiedad CSS. Estos elementos podemos declararlos de una forma extendida:

```
H1 {color: purple;}
H2 {color: purple;}
H3 {color: purple;}
H4 {color: purple;}
H5 {color: purple;}
H6 {color: purple;}
```

O utilizar la agrupación para conseguir una declaración más compacta (el resultado es el mismo en ambos casos):

```
H1, H2, H3, H4, H5, H6 {color: purple;}
```

Alineación

Nos permite alinear los textos dentro de elementos de bloque de las siguientes formas:

- Izquierda (left).
- Centrado (center).
- Derecha (right).
- Justificado (justify), es decir, centrado a derecha e izquierda.

Podemos utilizar el valor de "**text-align: center**" para reemplazar a la etiqueta CENTER que ya no se utiliza:

```
<html>
<head>
  <title>Propiedades del texto</title>
<style>
  h1 { text-align: left; }
  h2 { text-align: right; }
  h3 { text-align: center; }
</style>
</head>
<body>
<h1>Alineación izquierda</h1>
<h2>Alineación derecha</h2>
<h3>Alineación centrada</h3>
</body>
</html>
```

Transformación del texto

Permiten modificaciones en la presentación de las cadenas de texto mediante la propiedad "**text-transform**", la cual puede tomar alguno de los siguientes valores:

- **uppercase:** Todo el texto se transformará a mayúsculas.
- **lowercase:** Todo el texto se transformará a minúsculas.
- **capitalize:** Convierta a mayúscula la primera letra de cada palabra.
- **none:** No se realizará ninguna transformación sobre el texto.

```
<html>
<head>
<title>Propiedades del texto</title>
<style>
p { font-size: 200%; font-weight: bold;}
</style>
</head>
<body>
<p style="text-transform: uppercase;">
Prueba de transformación
</p>
<p style="text-transform: lowercase;">
Prueba de transformación
</p>
<p style="text-transform: capitalize;">
Prueba de transformación
</p>
</body>
</html>
```

PRUEBA DE TRANSFORMACIÓN

prueba de transformación

Prueba De Transformación

Decoración del texto

Añade a las cadenas de texto cierto formato mediante el uso de la propiedad "**text-decoration**", la cual puede tomar uno de los siguientes valores:

- **underline:** Subraya el texto asociado.
- **overline:** Dibuja un línea sobre el texto asociado. Es como un subrayado, pero sobre la parte superior del texto.
- **line-through:** Tacha el texto asociado dibujando una línea sobre el mismo.
- **blink:** Hace que el texto asociado parpadee.
- **none:** No aplica ningún elemento decorativo al texto.

```
<html> <head> <title>Propiedades del texto</title> <style>
p { font-size: 200%; font-weight: bold;}
</style>
</head>
<body>
<p style="text-decoration: underline;">
Prueba de decoración del texto
</p>
<p style="text-decoration: overline;">
Prueba de decoración del texto
</p>
<p style="text-decoration: line-through;">
Prueba de decoración del texto
</p>
<p style="text-decoration: blink;">
Prueba de decoración del texto (este está parpadeando :)
</p>
</body>
</html>
```

Prueba de decoración del texto

Prueba de decoración del texto

~~Prueba de decoración del texto~~

Prueba de decoración del texto

Clasificación de los elementos

En **CSS1**, los elementos formateados pueden agruparse en tres categorías:

- **Elementos de bloque.** Como los párrafos, los encabezados, las tablas, las listas, los DIV o el BODY. Son los que fuerzan un salto de línea al final de los mismos.
- **Elementos inline.** Como los enlaces, el énfasis o el SPAN. Pueden ser hijos de cualquier otro elemento.
- **Elementos de lista.** Son aquellos que en HTML sólo pueden contener elementos de tipo LI. Pueden ser automáticamente numerados o susceptibles de tomar algún tipo de estilo que se aplique a cada uno de los ítems que los definen.

Podemos cambiar la visualización de ciertos elementos mediante la propiedad **display**, la cual tiene las siguientes características:

- Puede tomar los valores: block, inline, list-item, none (oculta el elemento).
- Su valor por defecto es block.
- No se hereda.
- Es aplicable a todos los elementos.

Con display podemos cambiar aspectos típicos de visualización que están definidos en HTML, como el hecho de que los párrafos sean de tipo bloque:

P {display: inline;}

Pero cuando realmente se le saca partido a la propiedad display, es cuando formateamos un documento XML. Esto se debe a que XML no tiene un comportamiento predefinido para los elementos que contiene.

CSS2 completa los tipos de formateos añadiendo un tratamiento específico para tablas, filas y celdas.

Colores

- Lista de colores disponibles **por nombre**: aqua, gray, navy, silver, black, green, olive, teal, blue, lime, purple, white, fuchsia, maroon, red, yellow. Están tomados de los dieciseis colores básicos que se generan como VGA.
- Colores definidos **mediante valores RGB**. Pueden definirse como valores porcentuales entre 0 y 100 % (color: rgb(100%,80%,60%)) , o mediante valores numéricos entre 0 y 255 (color: rgb(255,0,0)).
- Colores definidos **mediante valores hexadecimales** con la estructura #RRGGBB (color: #FF0000). La notación hexadecimal puede abreviarse cuando sus valores se repiten para cada par, como en la expresión color: #000.
- **Colores web-safe.** Son aquellos cuya visualización se mantendrá entre navegadores. Existen 216 de estos colores, los cuales se obtienen mediante incrementos del 20%, 51 unidades o 33 en hexadecimal (rgb(40%,100%,80%), rgb(0,204,153) o #669933).

Estilo en los enlaces

Una técnica muy habitual, que se puede realizar utilizando las hojas de estilo en cascada y no se podía en HTML, es la definición de estilos en los enlaces, quitándoles el subrayado o hacer enlaces en la misma página con distintos colores.

Para aplicar estilo a los enlaces debemos definirlos para los distintos tipos de enlaces que existen (visitados, activos...). Utilizaremos la siguiente sintaxis, en la declaración de estilos global de la página (<STYLE>) o del sitio (Definición en un archivo externo):

Enlaces normales

A:link {atributos}

Enlaces visitados

A:visited {atributos}

Enlaces activos (Los enlaces están activos en el preciso momento en que se pincha sobre ellos)

A:active {atributos}

Enlaces hover (Cuando el ratón está encima de ellos)

A:hover {atributos}

El atributo para definir enlaces sin subrayado es text-decoration:none, y para darles color es color:tu_color.

También podemos definir el estilo de cada enlace en la propia etiqueta <A>, con el atributo style. De esta manera podemos hacer que determinados enlaces de la página se vean de manera distinta .

Ejemplo de estilos en enlaces :

```
<STYLE type="text/css">
A:link {text-decoration:none; color:#0000cc;}
A:visited {text-decoration:none; color:#ffcc33;}
A:active {text-decoration:none; color:#ff0000;}
A:hover {text-decoration:underline; color:#999999; font-weight:bold}
</STYLE>
```


Comentarios en HTML y CSS

En HTML definimos los comentarios de la siguiente forma:

```
<!-- Comentario en una página HTML -->
```

```
<!--  
Comentario  
en varias  
líneas  
-->
```

Los **comentarios en CSS** hacen uso de una sintaxis muy similar a la C/C++, es decir, se considerará un comentario cualquier sección del documento que comience con `/*` y termine con `*/`.

```
<STYLE type="text/css">  
/* Esto es un comentario */  
A:link {text-decoration:none;color:#0000cc;}  
A:visited {text-decoration:none;color:#ffcc33;}  
/*  
    Esto es un  
    comentario  
    de varias líneas  
*/  
</STYLE>
```

Estilos en los encabezados <H1> <H2> <H3>....

Los títulos también conocidos como encabezados, o headings en inglés, sirven para denotar importancia y jerarquía en una página o documento.

En HTML y/o XHTML, existen 6 niveles de encabezados `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`. Siendo el número 1 de mayor importancia y el 6 de menor importancia.

¿Por qué usarlos?

Por accesibilidad web, y facilidad para agregar estilos (CSS).

¿Qué diferencia hay entre un `<h1>` y un ``?

Son completamente diferentes, un encabezado es un encabezado, y cuando algún visitante entra en nuestra página con un celular, Pda o por Braille; estos dispositivos reconocen que este texto es un título, mientras que si fuera un texto con tamaño grande, los dispositivos no lo reconocerían y crearían que es un simple texto más, por tanto **se crearía una mala accesibilidad**.

Aunque sea un título de jerarquía 1 `<h1>`, éste puede tener un tamaño más pequeño gracias al CSS.

La etiqueta <embed>

```
<EMBED src="URL del fichero" width="500" height="500">
```

Con esta etiqueta podremos insertar cualquier tipo de fichero en nuestro documento: fichero de sonido, de vídeo, gráfico, etc. Como es lógico, el navegador deberá tener la capacidad de leer el tipo de fichero especificado o ayudarse de algún otro programa externo que le permita visualizar el fichero. En caso contrario dará un mensaje de error informándonos de que no puede leer el fichero especificado.

Con el argumento `src` indicamos la URL del fichero a mostrar. Con los argumentos `width` y `height` indicaremos el tamaño del objeto insertado, estos argumentos son opcionales.

Elementos para enfatizar

El elemento **strong** sirve para poner en negrita un texto.

Sus etiquetas son: `` `` o `` ``

El elemento **em** es el apropiado para marcar con énfasis las partes importantes de un texto.

Sus etiquetas son: `` y `` (ambas obligatorias)

```
<p>  
  <em>El dinero</em> es importante pero <strong>la salud</strong> más.  
</p>
```

em tiene un hermano mayor: **strong**.

em sirve para dar énfasis y **strong** para dar mucho énfasis.

Selector universal * (asterisco)

El selector universal se escribe con un asterisco (*) y representa a cualquier elemento de la página. Por ejemplo, con:

```
* {color: red}
```

Todos los elementos de la página tendrán como color de primer plano el rojo.

Resulta de utilidad cuando queremos inicializar todos los elementos de nuestra página con determinadas propiedades generales.

Estilos en las listas

Lista desordenada

```
<html>
<head>
  <style type="text/css">
    ul{list-style-type:square}
  </style>
</head>

<body>
  <p>Lista señalada con cuadrados</p>
  <ul>
    <li>ASP</li>
    <li>PHP</li>
    <li>ADO</li>
  </ul>
</body>
</html>
```

Lista ordenada

```
<html>
<head>
  <style type="text/css">
    ol{list-style-type:upper-roman}
  </style>
</head>

<body>
  <p>Lista ordenada con números romanos</p>
  <ol>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
  </ol>
</body>
</html>
```

La etiqueta <LINK>

```
<link rel="stylesheet" type="text/css" href="estilos.css" title="default">
```

Para conseguir la adecuada carga de la hoja de estilos, debemos posicionar este elemento en el HEAD de la página.

Atributos de la etiqueta LINK:

- REL. Define el tipo del documento o "con qué tiene relación". En general, su valor será "stylesheet".
- TYPE. Será siempre "text/css".
- HREF. Es el valor de la URL que nos permite el acceso al fichero de estilos.
- TITLE. Actualmente casi no se utiliza, pero será muy importante en el futuro. Su importancia radica en que puede haber en el mismo documento más de una etiqueta LINK. En el caso de que haya más de una etiqueta LINK, sólo aquellos cuyo valor de REL sea "stylesheet" se aplicarán inicialmente al documento:

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

El navegador cargará ambas hojas de estilo, combinando las reglas que definen y aplicando el resultado al documento.

Hojas de estilo alternativas

Es de especial utilidad hacer uso de la etiqueta TITLE, cuando definimos hojas de estilo alternativas. En el siguiente ejemplo se definen tres hojas de estilo dentro del mismo documento de forma que:

- La primera definición se aplicará siempre.
- El navegador nos permitirá seleccionar alguna de las dos siguientes como fuente alternativa de estilos. Además, cuando nos presente el menú para seleccionar la hoja de estilos alternativa, utilizará el atributo TITLE para su identificación.

```
<link rel="stylesheet" type="text/css" href="fija.css" title="default">  
  
<link rel="alternate stylesheet" type="text/css" href="alternativa1.css"  
title="Vista de impresion">  
  
<link rel="alternate stylesheet" type="text/css" href="alternativa2.css"  
title="Vista decorada">
```

La etiqueta <STYLE> y la directiva @import

La etiqueta **STYLE** es un elemento relativamente reciente en la especificación HTML. Permite dos tipos de tratamiento:

- Definir estilos que se aplicarán a la etiqueta en que se ubique, sin necesidad de definirlos en el HEAD del documento.
- Definir una hoja de estilo completa en el HEAD sin que esta se almacene en un fichero externo. Siempre debe hacer uso del atributo TYPE. Contiene estilos que se aplicarán al documento, pero también puede contener múltiples enlaces a hojas de estilo externas definidas con la directiva @import.

La directiva @import:

La única diferencia con la etiqueta LINK es su sintaxis y el lugar donde puede utilizarse. La directiva @import sólo puede utilizarse dentro de una región STYLE en el HEAD de la página. Además, estas directivas deben definirse antes de cualquier regla CSS que contenga esta región STYLE.

```
<style type="text/css">
  @import url(externo.css);
  h1 {color: gray;} </style>
```

Con la directiva @import no podemos definir hojas de estilos alternativas, y en algunos navegadores no se cachea estos contenidos.

Identificación y agrupación de elementos <CLASS> <ID>

¿Como se puede dar color a un título concreto de forma diferente a los otros títulos de tu sitio web? ¿Cómo se pueden agrupar los enlaces en diferentes categorías y dar a cada categoría un estilo especial? Estas dos preguntas son un ejemplo de lo que nos permiten hacer los atributos <id> y <class>.

¿Qué es el ID?

El atributo ID sería como el número de nuestra identificación personal o el número de pasaporte. En HTML el atributo ID indica el nombre único de un elemento. **Este nombre no puede repetirse en un documento HTML**, lo que quiere decir que no puede haber dos o más elementos con el mismo ID. Además cada elemento solo puede tener un ID.

Por ejemplo, muchos sitios web tienen un encabezado, un menú principal, un contenido y un pie de página, los cuales serían elementos únicos de una página. Si cada una de estas secciones están en un DIV y les asignáramos un ID tendríamos algo así:

```
<div id="encabezado"></div>
<div id="menu"></div>
<div id="contenido"></div>
<div id="pie-de-pagina"></div>
```

Otro ejemplo de la utilización del atributo ID podría ser el siguiente:

```
<h1 id="c1">Capítulo 1</h1>
...
<h2 id="c1-1">Capítulo 1.1</h2>
...
<h2 id="c1-2">Capítulo 1.2</h2>
...
<h1 id="c2">Capítulo 2</h1>
...
<h2 id="c2-1">Capítulo 2.1</h2>
...
<h3 id="c2-1-2">Capítulo 2.1.2</h3>
```

Si por ejemplo el capítulo 1.2 tuviese que estar en rojo, se podría hacer lo siguiente usando el siguiente código CSS:

```
#c1-2 {
    color: red;
}
```

¿Qué es el CLASS?

Podemos comparar el atributo CLASS con una clase universitaria donde cada estudiante es un elemento que comparten la misma clase, y a su vez, pueden haber estudiantes que tengan más de una clase. De manera similar, en HTML, el atributo CLASS asigna uno o varios nombres a un elemento que pueden repetirse. Un elemento HTML puede tener varios nombres definidos con este atributo.

Un uso común es cuando hacemos menús utilizando UL donde cada LI tiene un estilo gráfico definido a través de CSS, por ejemplo:

```
/* Código CSS */
a.rojo {
    color:red;
    text-decoration:none;
}
a.verde {
    color:green;
}

<!-- Código HTML -->
<a class="rojo" href="#">enlaces rojos</a><br>
<a class="rojo" href="#">enlaces rojos</a><br>
<a class="rojo" href="#">enlaces rojos</a><br>

<a class="verde" href="#">enlaces verdes</a><br>
<a class="verde" href="#">enlaces verdes</a><br>
<a class="verde" href="#">enlaces verdes</a><br>
```

En el ejemplo anterior vemos que los cuatros elementos LI son de la misma 'clase' (item). Sin embargo, el primer elemento LI pertenece también a una segunda 'clase' (first).

Diferencias en pocas palabras.

A este punto podemos decir que el atributo ID es único porque:

- Cada elemento solo puede tener un solo ID.
- Cada página solo puede tener un elemento con ese ID.

Y el atributo CLASS no es único porque:

- Varios elementos pueden tener la misma clase.
- Los elementos pueden tener varias clases diferentes.

En la declaración del atributo ID usamos el símbolo #

En la declaración del atributo CLASS utilizamos el símbolo . (punto).

```
/* Código CSS */
.estoEsUnaClase {
    color:red;
    text-decoration:none;
}

#yEstoUnID {
    color:green;
}
```

Otras cosas a tener en cuenta

Otro de los puntos que debemos saber es que si a un elemento le aplicamos un "ID" y un "class" el elemento heredará preferentemente las propiedades del "ID".
Por ejemplo:

```
.com{
    background:#939;
    font-size:18px;
}
#fil{
    background:#393;
    font-size:10px;
}
```

```
<div id="fil" class="com">
Este texto hereda las propiedades tanto del identificador como de la clase.
Pero si se repiten, toma como preferencia el del identificador.
</div>
```

La etiqueta

 sirve para aplicarle estilo a una pequeña parte de una página HTML. Por ejemplo, con ella podríamos hacer que una parte de un párrafo se coloree en rojo. Con no es habitual englobar un trozo muy grande de texto.

Ejemplo de la utilización de la etiqueta para colorear una parte de un texto:

```
<p>Hola, me llamo <span style="color:red">Josep Garcia</span>.</p>
```

La importancia de !important en css

!important funciona como una palabra clave para ignorar las reglas de cascada. Cualquier definición que vaya acompañada de un !important tendrá mayor importancia que cualquier otra.

Ahora veamos unos ejemplos para poder entenderlo bien:

Cuando tenemos una misma propiedad aplicada dos veces, el navegador hará caso a la última:

```
#main {
    width:600px;
    width:800px;
}
```

Por lo tanto en este ejemplo el navegador asignará 800 píxeles.

La declaración !important puede ser usada para dar prioridad a diferentes parámetros:

```
#main {
    width:600px !important;
    width:800px;
}
```

Ahora en este ejemplo se aplicará un ancho de 600 píxeles.

Internet Explorer 6 y versiones anteriores ignoraba esta palabra clave (!important) mientras que IE7 si la soporta por lo que se puede utilizar para realizar pequeños hacks CSS.

O lo que es lo mismo para poder aplicar diferentes diseños en una misma hoja de estilo tendremos que usar los Hacks, que no es otra cosa de utilizar reglas que únicamente las entiendan uno de los dos navegadores.

Por lo tanto lo podemos utilizar de tal manera que distingamos entre distintas reglas dependiendo del navegador y sin necesidad de usar Javascript, es un hack muy básico pero sin duda ayudará a entender para que sirve la palabra clave "!important":

```
#main {  
  margin: 0 auto 0;  
  max-width: 900px;  
  min-width: 770px;  
  width:auto !important;  
  width:800px;  
}
```

En este ejemplo aplica un ancho dinámico (auto) a todos aquellos navegadores que soportan "!important" mientras que aplica un ancho de 800 píxeles a aquellos que no lo soportan por ejemplo IE6.

Este pequeño truco no es realmente del todo útil pero sin duda te ayudará a comprender el significado de esta palabra clave CSS.

Pero no hay que olvidar que !important es muy potente porque se antepone al resto de reglas por lo que hay que usarla con cuidado como por ejemplo cuando queremos que una definición no sea sustituida por ninguna otra o para restablecer el valor de una propiedad que no nos acordamos donde le asignábamos un valor que ahora no queremos (sale algo en color rojo y no nos acordamos pues color: green !important; y listo).

¿Qué son las capas? La etiqueta <DIV>

Las capas no son más que unos recuadros, que pueden situarse en cualquier parte de la página, en los que podemos insertar contenido HTML. Dichas capas pueden ocultarse y solaparse entre sí, lo que proporciona grandes posibilidades de diseño.

Podemos insertar una capa a través de las etiquetas <div> y </div>, que como ya vimos, sirven para agrupar bloques de texto.

- A través del atributo **id** se le da un nombre a la capa, y a través del atributo **style** se establecen el resto de propiedades de la capa.

- A través de las propiedades **left** (izquierda) y **top** (superior) se establece la posición de la capa respecto a los márgenes izquierdo y superior de la página. Pueden tomar un número como valor, acompañado de **px** cuando haga referencia a píxeles, y acompañado de **%** cuando haga referencia a un porcentaje.

- A través de las propiedades **width** (anchura) y **height** (altura) se establece el tamaño de la capa. Pueden tomar un número como valor, acompañado de **px** cuando haga referencia a píxeles, y acompañado de **%** cuando haga referencia a un porcentaje.

La propiedad display

Puede tener diferentes valores:

```
div {  
  display: [ block | inline ];  
}
```

display:**block** fuerza un salto de línea encima y debajo del elemento en el que lo aplicas.

```
div {  
  display: block;  
}
```

display:block div 1

display:block div 2

Display:**inline** es justo lo contrario (aplicable a elementos que de por sí efectúan el salto de línea)

```
div {  
  display: inline;  
}
```

display:inline div 1

display:inline div 2

Para que la capa aparezca en la posición establecida, es necesario incluir también la propiedad **position** con el valor **absolute**. Si no se estableciera este valor, la capa se mostraría pegada al margen izquierdo, en la posición en la que hubiera sido insertada dentro del código.

La propiedad position

Tiene los siguientes valores posibles: **static**, **relative**, **absolute**, **fixed**.

```
div {  
  position: [ static | relative | absolute | fixed ];  
}
```

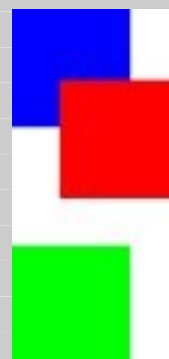
static

- Posición por defecto.
- La caja es una caja normal, situada de acuerdo al flujo normal.
- Las propiedades 'top', 'right', 'bottom' y 'left' no se aplican.

relative

- La posición de la caja se calcula de acuerdo al flujo normal (ésta se llama la posición en el flujo normal).
- Las propiedades top, right, bottom y left se aplican (arriba, derecha, abajo e izquierda). Estos desplazamientos son referentes a la posición natural de la caja en el flujo del documento (como hemos dicho en el punto anterior), y admiten tanto valores positivos como negativos. De manera que si colocamos una caja seguida de otra en el HTML, y por tanto esta segunda queda justo debajo de la primera en vertical, y quisiéramos que la segunda montara 40px sobre la primera, podríamos hacerlo de esta manera:

```
<!-- En HTML -->  
<div id="tres"></div>  
<div id="cuatro"></div>  
<div id="cinco"></div>  
  
<!-- CSS -->  
#tres, #cuatro, #cinco {  
  width: 100px;  
  height: 100px;  
}  
#tres {  
  background-color: blue;  
}  
#cuatro {  
  background-color: red;  
  position: relative;  
  top: -40px;  
  left: 40px;  
}  
#cinco {  
  background-color: green;  
}
```



absolute

- La posición de la caja se calcula respecto a las coordenadas de la página (no según el flujo normal).
- Las propiedades top, right, bottom, left se aplican.

Así, si queremos colocar una caja en las esquina inferior derecha de la pantalla, utilizaremos:

```
#derecha {position: absolute;
bottom: 0;
right: 0;
width: 50px;
height: 50px;
background-color: #f00;
}
```

con lo que la caja será colocada a 0px de la parte inferior y 0px de la derecha de la página.

- En el caso de que la caja esté dentro de otra las coordenadas del elemento en posición absoluta se medirán con respecto a la caja que lo contenga, **siempre que ésta a su vez tenga definida una posición relativa o absoluta**.

Hacemos hincapié porque esto es origen de muchas confusiones a la hora de maquetar. *Si tenemos una caja de 400px por 400px (por ejemplo), con posición relativa, e introducimos dentro de ella otra caja con posición absoluta y las coordenadas anteriores (bottom: 0; right: 0;), esta segunda caja se colocará en la esquina inferior derecha de la primera caja, no de la página completa, porque al estar dentro de una caja con posición relativa, las coordenadas de ubicación se tomarán con respecto a la primera caja (padre de la segunda). Sólo con quitar la posición relativa de la primera caja, la segunda saltaría a la esquina inferior derecha de la página.*

- Otra cosa a tener en cuenta de los elementos con posición absoluta es que no ocupan espacio físico real en el flujo del documento, así que, por un lado, una caja con esas características podrá ser colocada sobre cualquier otro elemento dentro del documento sin que éste se vea alterado en su posición, y por otro, si después de poner una caja con posición absoluta ponemos otra sin ella, esta segunda ocupará el espacio físico que le hubiera correspondido a la primera, ya que la primera no ocupa espacio, que queda disponible para otro elemento.

fixed

- La posición de la caja se calcula SIEMPRE respecto a las coordenadas de la página (aunque esté dentro de otra caja, siempre se coloca respecto a la página).
- Las propiedades top, right, bottom, left se aplican.
- Sirve para colocar una caja en la ventana y que no se mueva de su posición al hacer scroll.
- IE6 no entiende esta propiedad y tenemos que inventarnos una filigrana que nos permita simular el efecto "position:fixed" en IE6 y que sea compatible con todos los navegadores.

El truco está en ocultar el scroll de la ventana poniendo un "overflow: hidden" a los selectores de las etiquetas html y body (html, body {...}), crear un contenedor que ocupe el 100% de alto y ancho, que será el que tenga el scroll con "overflow: auto", y colocar con posición absoluta la caja que queríamos poner con posición fija.

```

/* Código CSS */

* {margin:0; padding:0;}
html, body {width:100%;
height:100%;
overflow:hidden;
}
#contenido {width:100%;
height:100%;
overflow: auto;
}
#fijo {
position:absolute;
width:200px;
height:200px;
background:#cff;
left:0;
top:0;
}

<!-- Código HTML -->
<div id="contenido">
  <p>Texto texto texto texto...</p>
</div>
<div id="fijo">Fijo</div>

```

La propiedad float

El efecto de esta propiedad es bastante simple pero resulta muy útil para que el diseño de la página se adapte al monitor del usuario. Lo que hace la propiedad es quitar al elemento del flujo normal de la página y colocarlo a la izquierda o derecha del resto del contenido de su elemento padre (lo hace "flotar" hacia la derecha o hacia la izquierda).

```

div {
  float: [ left | right | none ];
}

```

Por ejemplo, si quisiéramos texto con ajuste de línea alrededor de una imagen, el resultado sería el siguiente:



El código HTML del ejemplo anterior es el siguiente:

```
<!-- Código HTML -->
<div id="picture">
    
</div>
<p>causas naturales et antecedentes,
idcirco etiam nostrarum voluntatum...</p>

/* Código CSS */
#picture {
    float:left;
    width: 100px;
}
```

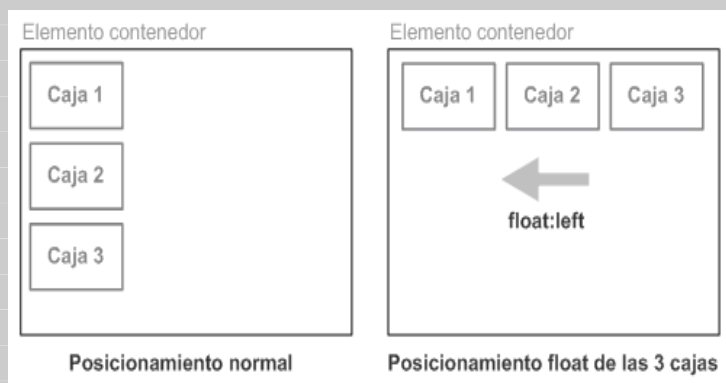
La propiedad float también se puede usar para **crear columnas en un documento**. Para crear dichas columnas tendrás que estructurar las columnas deseadas en el código HTML con la etiqueta <div>, como se muestra a continuación:

```
<!-- Código HTML -->
<div id="caja1">
    <p>Haec disserens qua de re agatur
    et in quo causa consistat non videt...</p>
</div>

<div id="caja2">
    <p>causas naturales et antecedentes,
    idcirco etiam nostrarum voluntatum...</p>
</div>

<div id="caja3">
    <p>nam nihil esset in nostra
    potestate si res ita se haberet...</p>
</div>

/* Código CSS */
#caja1 {
    float:left;
    width: 33%;
}
#caja2 {
    float:left;
    width: 33%;
}
#caja3 {
    float:left;
    width: 33%;
}
```



La propiedad clear

Esta propiedad indica cual de los lados de la caja de un elemento no puede quedar adyacente a una caja flotante anterior. Esto implica que la propiedad clear la especificamos sobre el siguiente elemento después de un elemento con float.

```
div {  
  clear: [ left | right | both | none ];  
}
```

left: El elemento en cuestión NO permite elementos flotantes a la izquierda.

right: El elemento en cuestión NO permite elementos flotantes a la derecha.

both: El elemento en cuestión NO permite elementos flotantes a ningún lado.

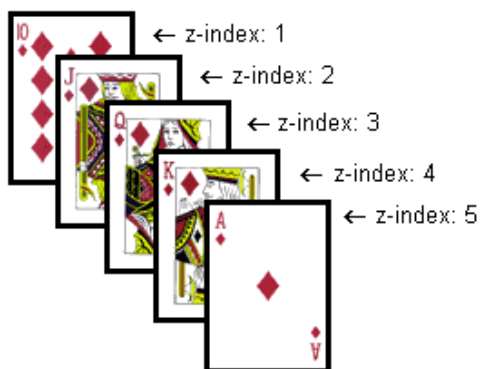
none: El elemento en cuestión permite elementos flotantes a ambos lados.

Capa sobre capa con la propiedad z-index

CSS funciona sobre tres dimensiones: altura, anchura y profundidad. Anteriormente nos hemos centrado en las dos primeras dimensiones. Para hacer referencia a la tercera dimensión (profundidad) utilizaremos la propiedad z-index, esta propiedad nos permite que los elementos se superpongan unos con respecto a otros.

Se puede asignar a cada elemento un número por medio de la propiedad z-index. El sistema consiste en que el elemento con un número mayor se superpone al elemento con un número menor.

Supongamos que estamos jugando al poquer y tenemos una escalera de color. La mano se puede presentar de tal manera que cada carta tiene un número asignado por medio de z-index:



En este caso, los números son consecutivos (yendo del 1 al 5), aunque se puede lograr el mismo resultado usando cinco números diferentes. Lo importante es la secuencia cronológica de los números (el orden).

El código del ejemplo de las cartas quedaría así:

```
#diez_de_diamantes {  
  position: absolute;  
  left: 100px;  
  top: 100px;  
  z-index: 1;  
}
```

```
#sota_de_diamantes {
    position: absolute;
    left: 115px; top: 115px;
    z-index: 2;
}
#reina_de_diamantes {
    position: absolute;
    left: 130px; top: 130px;
    z-index: 3;
}
#rey_de_diamantes {
    position: absolute;
    left: 145px; top: 145px;
    z-index: 4;
}
#as_de_diamantes {
    position: absolute;
    left: 160px; top: 160px;
    z-index: 5;
}
```

Ejemplos con capas

Ejemplo 1: Definir 3 elementos de tipo div, luego mediante la propiedad position desplazar el segundo div 15 píxeles a la derecha y 5 hacia abajo.

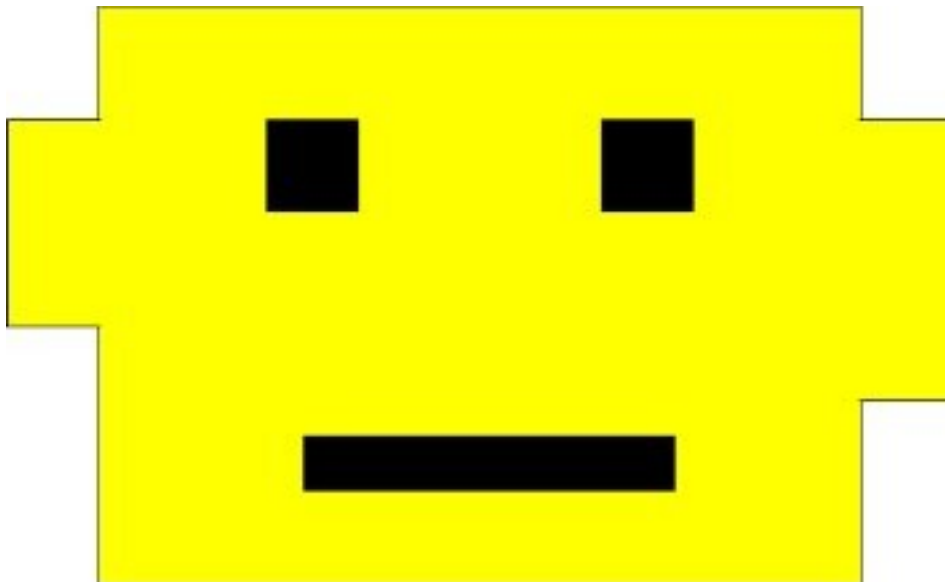
```
/* Código CSS */
#caja1,#caja2,#caja3 {
    background-color:#f99;
    font-family:verdana;
    font-size:1.3em;
}
#caja2 {
    position:relative;
    left:15px;
    top:5px;
}

<!-- código HTML -->
<div id="caja1">
<p>Esta es la primer caja, no se desplaza.</p>
</div>
<div id="caja2">
<p>Esta es la segunda caja. Se desplaza 15 píxeles a la derecha y 5 hacia
abajo de su posición por defecto.</p>
</div>
<div id="caja3">
<p>Esta es la tercer caja, no se desplaza.</p>
</div>
```


Ejemplo 2: Crear una cara con capas modificando únicamente la sección **style** del siguiente documento html.

```
<html>
<head>
  <style type="text-css">
  </style>
</head>
<body>
<div id="capa1">
  <div id="capa2"> </div>
  <div id="capa3"> </div>
  <div id="capa4"> </div>
  <div id="capa5"> </div>
  <div id="capa6"> </div>
  <div id="capa7"> </div>
</div>
</body>
</html>
```

Resultado:



Una de las posibles soluciones:

```
#capa1 {
  position: absolute;
  background-color: yellow;
  position: absolute;
  /*nos posicionamos en el centro del navegador*/
  top:50%;
  left:50%;
```

```

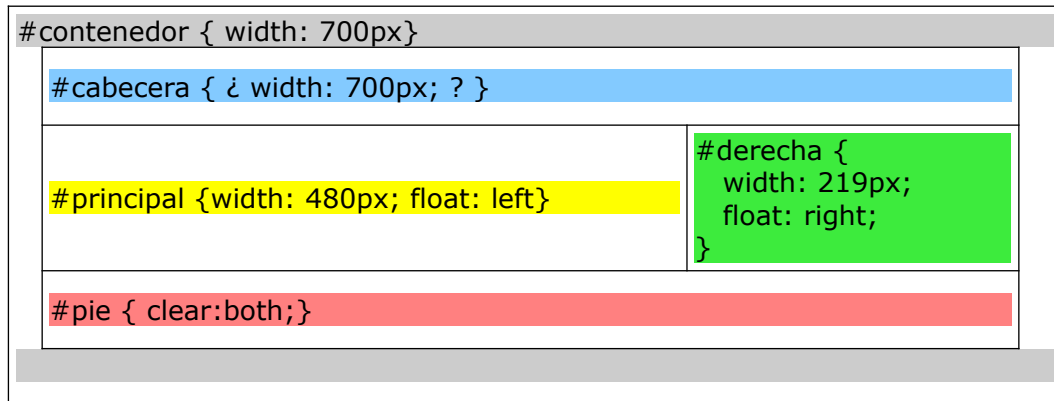
/*determinamos una anchura*/
width:400px;
/*indicamos que el margen izquierdo, es la mitad de la anchura*/
margin-left:-200px;

/*determinamos una altura*/
height:300px;
/*indicamos que el margen superior, es la mitad de la altura*/
margin-top:-150px;
border:1px solid #808080;
padding:5px;
}
#capa2, #capa3 {
    background-color: black;
    width: 50px; height: 50px;
    position: absolute;
    top:60px;
}
#capa2 { left:90px; }
#capa3 { right:90px; }

#capa4 {
    background-color: black;
    width: 200px;
    height: 30px;
    position: absolute;
    bottom:50px;
    right:100px;
}
#capa5, #capa6 {
    background-color: yellow;
    border: solid 1px;
    width:50px;
    height: 150px;
    position: absolute;
    top:60px;
}
#capa5 {
    right:-50px;
    border-left: 0px;
}
#capa6 {
    left:-50px;
    border-right: 0px;
}
}

```

Ejemplo 3: Crear una estructura de una web con una cabecera, un pie de página y 2 columnas (derecha y principal).



Creación de una página web

Pasos a seguir para crear una página web:

1. Tema de nuestro sitio (nombre de la empresa, sector....)
2. ¿Qué ofrecemos?
3. ¿A quién va dirigida nuestra página web?
4. Obtener información.
 - 10 páginas relacionadas o de la competencia.
 - Buscar información de nuestro producto.
5. Palabras clave que vamos a utilizar: keywords.
6. Escribir una descripción para nuestro sitio (2 líneas).
7. Buscar nombre de dominio según nombre de empresa o palabras clave.
8. Buscar alojamiento en un servidor (Hosting).
9. Empezamos a construir la web teniendo en cuenta:
 - Usar nombres de archivo (.html) que contengan palabras clave:
Si tu web vende fresas:
`http://dominio.com/producto.html` → mala elección
`http://dominio.com/venta-fresas.html` → buena elección
 - El archivo principal de la página (Inicio), tiene que llamarse obligatoriamente `index.html` o `index.htm`
10. Etiquetas META.
11. Mapa visual del sitio (para usuarios).
12. Mapa XML del sitio (para buscadores).
13. Robots.txt
14. Usabilidad y accesibilidad.
15. SEO
16. A tener en cuenta

¿ Qué son las palabras claves (Keywords) ?

Una Keyword es la palabra clave con la que nos llegaran las visitas desde los buscadores, por esta razón debemos pensar muy bien que Keywords vamos a utilizar.

Debemos buscar cuales son las mejores Keywords para nuestro web y para hacer eso podemos usar una poderosa herramienta de Google Adword llamada la "Herramienta de palabras claves":

<https://adwords.google.com/select/KeywordToolExternal>

Recomendaciones para las keywords de tu sitio:

- Introducir palabras clave con mala ortografía o que suelen ocurrir al teclear rápido. Por ejemplo "compar" (comprar).

Nombre de dominio

Para elegir un nombre de dominio tenemos dos opciones:

- 1- Contratarlo en una empresa diferente de la que hayamos elegido para el alojamiento.
- 2- Contratar el dominio y el hosting en la misma empresa.

La diferencia es que si contratamos el dominio y el hosting en empresas diferentes, tendremos que configurar nuestro dominio (desde el panel de control del dominio) cambiando las DNS por las que nos ofrece la empresa de hosting (esto puede tardar unas 48 horas en actualizarse) para que funcione correctamente.

Si contratamos el dominio y el alojamiento en la misma empresa no hace falta configurar nada y en unas horas lo tendremos todo funcionando.

Es recomendable buscar un nombre de dominio que esté relacionado con el nombre de la empresa o con algunas de las palabras clave que hayamos elegido.

Para saber si un dominio está libre se utilizan herramientas llamadas "WHOIS", una de las más conocidas es:

<http://www.directnick.com/whois/>

(En todos los sitios donde vayamos a contratar un dominio está disponibles este tipo de herramientas "WHOIS").

Alojamiento (hosting)

Cualquier servicio que se quiere ofrecer en Internet tiene que brindarlo un servidor, que es un ordenador que se encuentra encendido las 24 horas del día y conectado a Internet también permanentemente. En el caso de una página web, existen unos servidores que son los encargados de mandarla cuando se la solicita, son los servidores web. Nuestras páginas tienen que estar alojadas en un servidor web para que puedan estar accesibles desde Internet. Dependiendo del proyecto que estemos realizando necesitaremos más o menos recursos de nuestro servidor web. Si estamos haciendo una página personal quizás este punto no nos interese mucho y simplemente no queramos gastar dinero, pero si estamos pensando en hacer un proyecto más serio, como una página para nuestro negocio, conviene contratar un buen servicio de alojamiento de la página. No merece la pena realizar un esfuerzo grande en construir el sitio y luego colocarlo en un servidor de poca calidad, que sea lento o sufra cortes de servicio habitualmente.

Para elegir un alojamiento (hosting) para nuestra página web debemos tener en cuenta:

Espacio en el disco duro: El tamaño (peso) máximo, en Mb o Gb, de archivos que podremos subir al servidor.

Transferencia mensual: Cantidad de megas de transferencia mensual que el proveedor permite realizar desde nuestro dominio hacia fuera. Es decir, las páginas web que manda el servidor tienen un peso en Kbytes y el proveedor los va contando, cuando pasamos el límite que le han asignando al dominio nos cobran el exceso según un precio.

Velocidad del servidor: Ten en cuenta que si estás en España, un servidor situado en tu país será más rápido para ti, en la misma condición de líneas, que uno que esté en Estados Unidos. Cuanto más lejos tenga que viajar la información, tarda más. Pero un detalle, si un usuario de Colombia visita una página que está en un servidor español, para él tu servidor será más lento que uno de su país.

Bases de datos: En principio no necesitamos ninguna base de datos para nuestras páginas web, pero más adelante quizás queramos instalar algún script php que necesite 1 base de datos MySQL.

Características HW: Memoria RAM, Microprocesador, discos duros, línea utilizada. Esta información no la suelen dar en las empresas de hosting.

Características SW: soporte php, asp

Sistema operativo del servidor: recomendado hosting LINUX.

Precio €: precio mensual/anual del servidor, aquí hay que ir con cuidado porque en muchas empresas de hosting nos dan un precio bastante económico que lleva al lado un asterisco (*), este asterisco significa normalmente que el precio es el que nos indican pero si contratamos el alojamiento para X años. (Tal vez a nosotros nos interese únicamente contratarlo para un año).

Reputación de la empresa: buscar información en los foros.

Extras: Número de direcciones de correo personalizadas....

Etiquetas META

Las metatags son etiquetas html que **se incorporan en el encabezado de una página web (entre <head> y </head>)** y que resultan invisibles para un visitante normal, pero de gran utilidad para navegadores u otros programas que puedan valerse de esta información.

Su propósito es el de incluir información (metadatos) de referencia sobre la página: autor, título, fecha, palabras clave, descripción, etc.

Esta información podría ser utilizada por los robots de búsqueda para incluirla en las bases de datos de sus buscadores y mostrarla en el resumen de búsquedas o tenerla en cuenta durante las mismas y será invisible para un visitante normal.

Estas etiquetas también se usan para especificar cierta información técnica de la cual pueden valerse los navegadores para mostrar la página, como el grupo de caracteres usado, tiempo de expiración del contenido, posibilidad de dejar la página en cache o calificar el contenido del sitio ("para adultos", "violento"...)....

Sintaxis:

```
<meta name=" " content=" " />
```

Para mejorar la posición en buscadores utilizamos las siguientes:

```
<html>
<head>
<title>Título del sitio</title>
<meta name="keywords" content="keyword1,keyword2,keyword3,kw4..." />
<meta name="description" content="Descripción de la página web" />
</head>
<body> </body>
</html>
```

Mapa visual del sitio (para usuarios)

Nombre del archivo: mapa-web.html

Archivo html que contiene enlaces a todas las páginas que forman nuestro sitio web:

```
<a href="http://dominio.com/venta-fresas.html">Venta de fresas</html>
<a href="http://dominio.com/contacto.html">Contacte con nosotros</html>
.....
.....
```

Mapa XML del sitio (para buscadores)

Es un mapa del sitio en formato XML, donde pueden encontrarse fácilmente todas las páginas del sitio web, incluso aquellas antiguas y ya olvidadas, que no se llegarían a indexar si no encuentra un enlace apuntando hacia ellas.

Nombre del archivo: sitemap.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://www.dominio.com/</loc>
    <lastmod>2009-09-05</lastmod>
    <changefreq>weekly</changefreq>
    <priority>1</priority>
  </url>
  <url>
    <loc>http://www.dominio.com/venta-fresas.html</loc>
    <lastmod>2009-09-05</lastmod>
    <changefreq>weekly</changefreq>
    <priority>0.9</priority>
  </url>
  <url>
    ..... todas las direcciones de los archivos de la web
  </url>
</urlset>
```

lastmod: última modificación (poner fecha actual).

changefreq: frecuencia con la que se actualiza nuestra página web (dejar weekly por defecto).

priority: prioridad de cada una de las diferentes secciones, si vendemos naranjas le pondremos prioridad 1 al archivo venta-de-naranjas.html y prioridad 0.3 (por ejemplo) al archivo recetas-con-naranjas.html

Una vez creado el **sitemap.xml** lo podemos mandar a Google para que nos lo indexe en:

<http://www.google.com/webmasters/sitemaps/login>

Robots.txt

Ejemplo de fichero <http://dominio.com/robots.txt>

```
User-agent: *  
Allow: /  
Disallow: /directorio_privado/  
Sitemap: http://www.dominio.com/sitemap.xml
```

User-agent: Todos los bots de los buscadores.

Allow: Permitir que se rastree todo el sitio.

Disallow: No examinar el directorio_privado (contiene información sensible).

Sitemap: URL del sitemap.xml

Usabilidad y Accesibilidad

Usabilidad es un atributo de calidad que mide lo fáciles de usar que son las interfaces web.

Nielsen (2003)

Accesibilidad es la posibilidad de que el mayor número posible de personas pueda entrar a una web y usarla, indiferentemente de las limitaciones propias del individuo o de las derivadas del contexto de uso. Hassan y Martín.

Test de accesibilidad: <http://www.tawdis.net/>

Reglas para crear una web **usable**.

- Rápido

Un site solo capta la atención de un usuario durante los primeros 8 segundos que el usuario esta delante de la página web, pasado este tiempo, si el usuario no encuentra la información que esta buscando, cancelará y se ira a otro web.

- Simple

Mantenga una navegación constante. No fuerce a los visitantes a aprender diversos caminos o esquemas para la navegación en diversas partes de su site.

- Investigable

Los motores de búsqueda buscan el texto real. No prestan ninguna atención a los gráficos (incluso gráficos que parecen texto) y al código de programación (como el Javascript, usado para los menús y otros efectos especiales). Evite estas situaciones si desea que su web este bien posicionada en los buscadores.

- Para la mayoría

Los sites necesitan ser compatibles con todos los navegadores y ordenadores.

- Actualizado

La manera más rápida para que un web pierda credibilidad es contener la información anticuada. Incluso cosas pequeñas como una fecha del copyright de "2000", etc, pueden dañar la credibilidad del web además de su contenido.

Un poco de SEO (Search Engine optimization)

La tarea de ajustar la información de las páginas que se pretenden hacer aparecer en primeras posiciones de los resultados es conocida como SEO, sigla en inglés de Search Engine Optimization, optimización para motores de búsqueda. Consiste en aplicar diversas técnicas tendientes a lograr que los buscadores de Internet sitúen determinada página web en una posición y categoría alta (primeras posiciones) dentro de su página de resultados para determinados términos y frases clave de búsqueda. También es entendido como las técnicas de desarrollo web que tengan como objetivo mejorar la posición de un determinado sitio web por sus páginas en la lista de resultados de los Motores de Búsqueda que en inglés se identifican como SERPs y forma parte de la jerga de los webmasters hispanohablantes.

La aplicación de técnicas SEO es intensa en sitios web que ofertan productos o servicios con mucha competencia, buscando el posicionar determinadas páginas sobre las de sus competidoras en la lista de resultados (SERPs).

Las técnicas SEO pueden ser desmedidas y afectar los resultados naturales de los grandes buscadores por lo que si incumplen las cláusulas y condiciones de uso de los mismos pueden ser consideradas, en algunos casos, como una forma de SPAM, el spamdexing.

El trabajo es amplio, ya que el posicionamiento involucra al código de programación, al diseño y a los contenidos. También nos referimos a SEO para definir las personas que realizan este tipo de trabajo.

- Conseguir que otras webs de temática relacionada enlacen con tu web
- Darse de alta en directorios importantes como [Dmoz](#), [Yahoo!](#).
- Registrarse y participar en foros, de preferencia en foros temáticos relacionados a la actividad de su página web (añadiendo en la firma un enlace a nuestro sitio).
- Hay cientos de redes sociales, por ejemplo Hi5, Facebook, Orkut, entre cientos de comunidades en las cuales poder participar y obtener visitas de nuestros nuevos "amigos".
- Crear contenidos con textos que contienen frases con que buscan los usuarios web los servicios y/o productos de este sitio. Para hacer la investigación de las frases es importante de usar la base de datos de los buscadores como [Overture](#).
- Hacer nuestra web lo más accesible posible: limitar contenido en flash, frames, javascript...
- Crear títulos únicos y descripciones pertinentes del contenido de cada página.
- Enlazar internamente las páginas de nuestro sitio de manera ordenada y clara.
- Actualizar la página con contenido original de calidad.

Optimización de la web para buscadores

Pasos a seguir para optimizar nuestra web:

- Utilizar palabras clave con mala ortografía o resultantes de haber tecleado rápidamente, por ejemplo "COMPAR" (comprar).
- Usar nombres de archivo (.html) que contengan palabras clave:
Si tu web vende fresas:
`http://dominio.com/producto.html` → mala elección
`http://dominio.com/venta-fresas.html` → buena elección
- No usar guiones bajos en los nombres de archivo, usar guiones medios (normales).
`http://dominio.com/venta_de_fresas.html` → mal
`http://dominio.com/venta-de-fresas.html` → buena elección
- Usar palabras clave en el título de la página.
`<head><title>Venta de fresas y fresones valencia</title></head>`
- Usar título y etiquetas "META" diferentes para cada página.
- Usar cabeceras (**h1, h2, h3...**) para definir importancia en el contenido y colocar en ellas las palabras clave.
- Rellenar todas las etiquetas "**alt**" y "**title**" de los enlaces e imágenes con palabras clave.
- Servirnos de la mejor tecnología para que la página cargue rápido y el usuario no tenga que esperar (optimizar imágenes, archivos css, código html...)
- No utilizar **frames** de ningún tipo.
- Imprescindible crear un mapa visual para el usuario (.html).
- Imprescindible crear un mapa visual para buscadores (.xml).
- Imprescindible crear el archivo **robots.txt** para buscadores.

A tener en cuenta

Uso de imágenes: Otro tema que nos inquieta es el uso de las imágenes, ¿se justifica realmente su uso en Internet? Por lo general una imagen la usamos para acrecentar el impacto visual en el sitio, pero hay veces que, mal usada, lo único que acrecienta es nuestra espera para que aparezca en pantalla, ya que es uno de los elementos que incrementan el peso de una página web. Debemos pensar dos veces, si realmente se justifica el uso de una imagen en el documento que está creando.

Fuentes

- <http://www.josepgarcia.com>
- <http://www.manualdecss.com/>
- <http://mariuzzo.com/2009/06/18/diferencias-entre-atributos-id-y-class/>
- <http://www.desarrolloweb.com>
- <http://www.virtualnauta.com/es/css/csslistas.php>
- <http://www.w3c.es/Divulgacion/GuiasReferencia/CSS21/>
- <http://es.html.net/tutorials/css/>
- <http://willxd.com/diferencias-entre-id-y-class/>
- <http://www.aulaclie.es/html/>
- <http://www.carlosleopoldo.com/post/entendiendo-el-posicionamiento-con-css/>
- <http://www.araudi.net/forosdelweb/posiciones.html>

Reglas de usabilidad:

http://www.webtaller.com/maletin/articulos/reglas_de_usabilidad_web.php

Posicionamiento en buscadores

http://es.wikipedia.org/wiki/Posicionamiento_en_buscadores

Posicionamiento de los elementos (divs)

<http://es.html.net/tutorials/css/lesson14.asp>

Propiedad profundidad en las capas (z-index)

<http://es.html.net/tutorials/css/lesson15.asp>

Construcción de páginas web

<http://www.desarrolloweb.com/articulos/alocar-las-paginas.html>

Ejemplos CSS

<http://www.cssplay.co.uk/menu/>

Ejercicios CSS

<http://www.librosweb.es/css/capitulo15.html>

La importancia de !important en CSS

<http://www.tufuncion.com/important>

Css Sprites

<http://www.tufuncion.com/css-sprites>

SEO

<http://www.ciberwolf.com/2009/01/seo-para-tontos-parte-1-que-son-las-keywords-cuales-son-las-que-atrapan-mas-visitas-y-las-mejores-pagadas-en-adsense.html>

CSS templates

- Sencillos

<http://www.code-sucks.com/css%20layouts/fixed-width-css-layouts/>

<http://www.code-sucks.com/css%20layouts/faux-css-layouts/>

<http://www.jimmylin.info/templates.html>

http://www.thenoodleincident.com/tutorials/box_lesson/boxes.html

- Más "profesionales"

<http://www.solucija.com/free-templates>

<http://www.free-css-layouts.com/free-layouts/>

<http://www.opensourcetemplates.org/browse/>

<http://templates.arcsin.se/category/website-templates/>

http://www.ex-designz.net/template/tempcat.asp?cat_id=13

<http://www.template.com/page/1>

http://www.oswt.co.uk/browse_designs.html

Otros recursos

¿Cómo centrar un <DIV>?

<http://www.lawebdelprogramador.com/codigo/codigo.php?idp=1587&id=268&texto=CSS>

<http://micodigobeta.com.ar/?p=340>

```
#capal {
    position: absolute;
    background-color: yellow;
    position: absolute;
    /*nos posicionamos en el centro del navegador*/
    top:50%;
    left:50%;

    /*determinamos una anchura*/
    width:400px;
    /*indicamos que el margen izquierdo, es la mitad de la anchura*/
    margin-left:-200px;

    /*determinamos una altura*/
    height:300px;
    /*indicamos que el margen superior, es la mitad de la altura*/
    margin-top:-150px;
    border:1px solid #808080;
    padding:5px;
}
```

Errores, sugerencias

josepcarlet@Gmail.com

Resumen CSS

Fuentes

color	Color del texto, lo admiten casi todas las etiquetas HTML.	<color>
font-family	Sirve para indicar el tamaño de las fuentes de manera más rígida y con mayor exactitud.	Arial sans-serif cursive fantasy monospace (Todas las fuentes habituales)
font-style	Es el estilo de la fuente	[normal italic oblique]
font-weight	Intensidad de la fuente	[normal bold bolder lighter 100 200 300 400 500 600 700 800 900]
font-size	Tamaño de la fuente	[[xx-small x-small small medium large x-large xx-large] [larger smaller] <longitud> <porcentaje>]

Texto

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

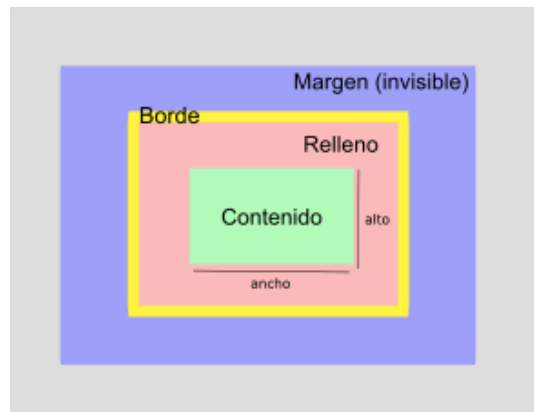
Listas

list-style-type	Estilo aplicable a los marcadores visuales de las listas	[disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian lower-alpha upper-alpha none]
list-style-image	Imagen aplicable a los elementos de las listas	[url("http://...") none]
list-style-position	Posición dentro de la lista de los elementos marcadores de las listas	[inside outside]

Fondo

background-color	Color de fondo	[<color> transparent]
background-image	Imagen de fondo	[url(...) none]
background-repeat	Repetición de la imagen de fondo	[repeat repeat-x repeat-y no-repeat]
background-attachment	Desplazamiento de la imagen de fondo	[scroll fixed]
background-position	Posición de la imagen de fondo	[[<porcentaje> <longitud> left center right] [<porcentaje> <longitud> top center bottom]?] [[left center right] [top center bottom]]

Modelo de cajas



Márgenes		
margin	Ancho para varios márgenes individuales	[<longitud> <porcentaje> auto]{1,4}
margin-top margin-right margin-bottom margin-left	Tamaño del margen superior, derecho, inferior e izquierdo.	[<longitud> <porcentaje> auto]
Relleno		
padding	Tamaños para varios rellenos individuales	[<longitud> <porcentaje>] {1,4}
padding-top padding-right padding-bottom padding-left	Ancho del relleno superior, derecho, inferior e izquierdo	[<longitud> <porcentaje>]
Bordes		
border-width	Ancho del borde	[thin medium thick <longitud>] {1,4}
border-top-width border-right-width border-bottom-width border-left-width	Anchura del borde superior, derecho, inferior o izquierdo	[thin medium thick <longitud>]
border-color	Color del borde	[<color> transparent] {1,4}
border-top-color border-right-color border-bottom-color border-left-color	Color del borde superior, derecho, inferior o izquierdo	[<color> transparent]
border-style	Estilo del borde	[none hidden dotted dashed solid double groove ridge inset outset] {1,4}
border-top-style border-right-style border-bottom-style border-left-style	Estilo del borde superior, derecho, inferior o izquierdo	[none hidden dotted dashed solid double groove ridge inset outset]