



UNIDAD 1_2_USO DE ESTILOS

1 INTRODUCCIÓN AL USO DE ESTILOS

1.1.INTRODUCCIÓN CSS

1.2.PREFERENCIA DE APLICACIÓN DE ESTILOS

1.3.CONCEPTOS DE MARGEN, PADDING Y BORDE

1.4.CSS NIVEL 3

1.5.SELECTORES

1.5.1 SELECTORES DE ATRIBUTO

1.5.2 SELECTORES GENERALES DE HERMANOS

1.5.3 SELECTORES PSEUDO-ELEMENTOS

1.5.4 SELECTORES PSEUDO-CLASES

1.5.5 OTROS SELECTORES

2 COLOR

2.1.COLOR INTRODUCCIÓN

2.2.COLOR. FORMATOS DE COLOR

2.3.COLOR. EJEMPLOS

2.4.COLOR TRANSPARENCIA

2.5.COLOR. TRANSPARENCIA EJEMPLOS

3 UNIDADES DE MEDIDA

3.1.UNIDADES ABSOLUTAS

3.2.UNIDADES RELATIVAS

3.3.PORCENTAJES

3.4.RECOMENDACIONES

4 EFECTOS CSS

4.1.SOMBRAS

4.2.FONDOS

4.3.GRADIENTES

4.4.BORDES

4.5.TEXTO

4.5.1 EFECTOS DE TEXTO

4.6.TRANSFORMACIONES

4.7.TRANSICIONES Y ANIMACIONES

4.7.1 TRANSICIONES

4.7.2 ANIMACIONES

1 INTRODUCCIÓN AL USO DE ESTILOS

1.1.INTRODUCCIÓN CSS

En un principio, los documentos Web sólo permitían el uso del lenguaje de etiquetas HTML. Cuando era necesario modificar alguna característica de formato se utilizaba etiquetas o atributos HTML orientados a cambiar el tamaño del texto (<h1>, <h2>, <h3>, etc), el color (), la alineación (<center>), etc.

En diciembre de 1996 el W3C publicó la primera recomendación de **Hojas de Estilo en Cascada** o **CSS** (Cascading Style Sheets) con el objetivo de dotar a la Web de un conjunto de técnicas específicas para el formato. Al fin era posible fijar el color, el tipo de letra y otras características de visualización con un lenguaje propio y se alcanzaba el objetivo buscado de separar la estructura de un documento de su presentación.

En la actualidad, podemos encontrar nuevas especificaciones de CSS3 en la siguiente dirección <https://www.w3.org/Style/CSS/specs.en.html>

1.2.PREFERENCIA DE APLICACIÓN DE ESTILOS

Si se define la misma propiedad para la misma etiqueta con el mismo selector en dos sitios distintos, las reglas de precedencia son las siguientes:

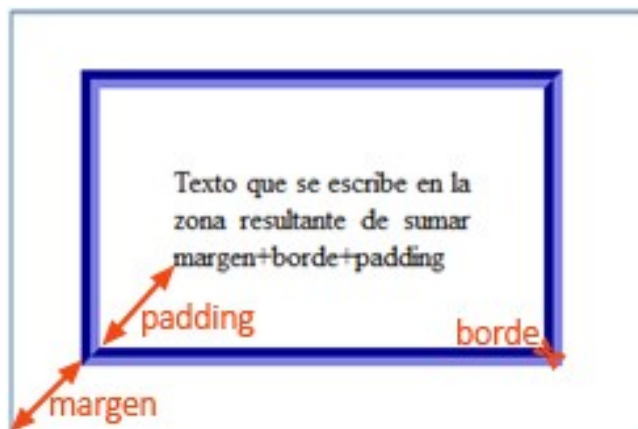
- Las propiedades definidas en un atributo style se imponen a las propiedades definidas en la etiqueta <style>.
- Las propiedades definidas en la etiqueta <style> se imponen a las propiedades definidas en una hoja de estilo enlazada.
- Las propiedades definidas en un atributo style se imponen a las propiedades definidas en una hoja de estilo enlazada.

Además de estas propiedades definidas por el creador de la página web, hay que tener en cuenta que también se aplican las propiedades definidas en la hoja de estilo por defecto del navegador.

Si las propiedades se encuentran definidas en diferentes hojas de estilo, el navegador aplica el valor definido en la última hoja de estilo enlazada (es decir, en el último enlace <link> del <head>).

1.3.CONCEPTOS DE MARGEN, PADDING Y BORDE

En un contenedor (*body*, *div*, *section*, etc.) se pueden establecer varias zonas diferenciadas que servirán de puntos de referencia para múltiples operaciones CSS. Estas zonas vienen delimitadas por el *margen* y el *padding*. El *margen* es la distancia entre el extremo del contenedor y su contenedor padre en el DOM y el *padding* es la distancia existente entre el borde establecido y la zona de contenido. Podemos ver un esquema de estos bloques en la siguiente imagen.



En CSS existen propiedades para establecer el margen (*margin*), el borde (*border*) y el padding (*padding*). Por ejemplo, el siguiente código establece una capa con margen 30 píxeles y padding 40 píxeles, el borde será de una anchura de 10 píxeles con efecto de tres dimensiones y color azul.

```
<style>

div{width:160px;

    margin:30px;

    border:10px groove #00C;

    padding:40px;

}

</style>
```

Propiedades como margin, border o padding se complementan con otras que permiten desglosar sus características, así podemos determinar las medidas por la izquierda (left), por la derecha (right), por arriba (top) y por abajo (bottom).

1.4.CSS NIVEL 3

Las distintas recomendaciones de CSS realizadas por el W3C no se denominan versiones sino niveles, así se habla de CSS nivel 1 (1996), CSS nivel 2 (1998) (con una revisión denominada nivel 2.1) y CSS nivel 3 (en desarrollo). Cada nivel pretende mejorar el nivel anterior además de incluir todas sus normas.

Alguno de los módulos que el W3C está desarrollando son:

- **Módulo de color:** Permite utilizar propiedades relacionadas con el color del texto, fondos, bordes y otras partes de elementos en un documento.
- **Módulo de composición para plantillas.** Dispone de funciones de posicionamiento y alineación de elementos.
- **Módulo de marquesina.** Permite al desarrollador realizar un efecto marquesina para mostrar un contenido que no cabe entero en un contenedor.
- **Módulo de diseño multi-columna.** Describe los métodos para que el texto se muestre distribuido en columnas.
- **Módulo de fondos y bordes** Contiene los métodos que permiten actuar sobre los fondos y bordes de los distintos elementos.

El grado de soporte de los navegadores para el último nivel de las hojas de estilo es muy variable. El soporte de CSS3 depende del motor que utilice cada familia de navegadores, por ello es necesario conocer que motor utiliza cada navegador.

Se puede conocer el grado de desarrollo de cada uno de los módulos consultando la página <http://www.w3.org/Style/CSS/current-work>

1.5.SELECTORES

1.5.1 SELECTORES DE ATRIBUTO

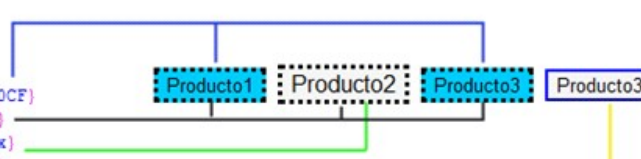
Todas las etiquetas HTML admiten el uso de múltiples atributos. En función de estos atributos se pueden crear selectores para aplicar estilos a las etiquetas que los posean.

- **elemento[atributo].** Selecciona los elementos que contengan el atributo indicado independientemente de su valor.

- **elemento[atributo=valor]**. Selecciona los elementos que contengan el atributo indicado con un determinado valor.
En CSS3, los selectores de atributo admiten el uso de filtro como podemos observar a continuación:
- **elemento[atributo="texto"]**. Selecciona aquellos elementos que contienen el atributo que se indica con el valor "texto".
- **elemento[atributo^="texto"]**. Selecciona aquellos elementos que contienen el atributo indicado y cuyo valor comienza por la subcadena "texto".
- **elemento[atributo\$="texto"]**. Selecciona aquellos elementos que contienen el atributo indicado y cuyo valor finaliza con la subcadena "texto".
- **elemento[atributo*="texto"]**. Selecciona aquellos elementos que contienen el atributo indicado y cuyo valor contiene la subcadena "texto".

En el siguiente ejemplo podemos ver el efecto de utilizar distintos selectores de atributo sobre distintos botones.

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <style>
    button[data-precio^="100"]{background:#0CF}
    button[data-precio$="€"]{ border:dotted}
    button[data-precio*="5"]{ font-size:16px}
    button[data="sumar"]{ border-color:#00F}
  </style>
</head>
<body>
  <button data-precio="100€" id="uno">Producto1</button>
  <button data-precio="50€" id="dos">Producto2</button>
  <button data-precio="1000€" id="tres">Producto3</button>
  <button data="sumar" id="comprar">Comprar</button>
</body>
</html>
```



1.5.1 SELECTORES GENERALES DE HERMANOS

- **elemento1~elemento2**. Selecciona el **elemento2** que es hermano de **elemento1** siempre que en el código HTML esté primero el **elemento1** y después **elemento2**, pudiendo haber entre ellos otras etiquetas.
El combinador ~ separa dos selectores y selecciona el segundo elemento sólo si está precedido por el primero y ambos comparten un padre común.
- **elemento1+elemento2**: Se hace referencia a este selector como selector adyacente o selector del próximo hermano. Sólo seleccionará un elemento especificado que esté inmediatamente después de otro elemento especificado.

En la siguiente imagen se muestra un ejemplo de utilización de selectores de elementos adyacentes (+) y de selectores de hermanos (~).

En este caso el primer selector **h1+h2** sólo afecta a la etiqueta <h2> que está inmediatamente después de <h1>, en cambio el selector **h1~h2** afecta a aquellas etiquetas <h2> que están en el mismo nivel y después de <h1>, pudiendo haber otras etiquetas entre ellas. Cabe destacar que la etiqueta <h2> que está dentro de <section> no es seleccionada pues no es hermana de <h1>.

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <style>
    h1~h2 {color:#3CF;}
    h1+h2 {border:dashed;}
  </style>
</head>
<body>
  <header>
    <h1>Escucha y sorpréndete</h1>
    <h2>La prensa</h2>
  </header>
  <article>
    <h1>Noticia de hoy</h1>
    <p>Todas las noticias curiosas del planteta</p>
    <section>
      <h2>¡Sorpréndete!</h2>
    </section>
    <h2>Un perro en la cocina</h2>
  </article>
</body>
</html>
```

Escucha y sorpréndete

¡La prensa

Noticia de hoy

Todas las noticias curiosas del planteta

¡Sorpréndete!

Un perro en la cocina

1.5.2 SELECTORES PSEUDO-ELEMENTOS

Los pseudo-elementos son filtros de selección que se pueden aplicar a las distintas etiquetas HTML. Los pseudo-elementos de CSS 2.1 forman parte de CSS3 pero cambian su sintaxis ya en lugar de utilizar únicamente un único símbolo de dos puntos (:), lo utilizan doble (::), por ejemplo:

- **elemento::first-line**, selecciona la primera línea del texto de un elemento
- **elemento::first-letter**, selecciona la primera letra del texto de un elemento
- **elemento::before**, selecciona la parte anterior al contenido de un elemento para insertar nuevo contenido generado
- **elemento::after**, selecciona la parte posterior al contenido de un elemento para insertar nuevo contenido generado
- CSS3 aporta un nuevo selector de pseudo-elementos
- **elemento::selection**, hace referencia al texto que el usuario ha seleccionado con el ratón o el teclado
- **elemento::-moz-selection**: es el selector utilizado para los navegadores del motor Gecko

En la siguiente imagen se puede observar el efecto de utilizar los pseudo-elementos descritos en el navegador Safari. Cabe destacar que la zona que se muestra amarilla está seleccionada con el ratón desde el navegador y que la propiedad **content** inserta texto en el lugar seleccionado.

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <style>
    hl::before {content: 'Poesía: '}
    p::first-letter {font-size:36px}
    p::first-line{font-style:italic}
    p::selection{background-color:#FF0}
    p::after{content: 'Miguel Hernández'}
  </style>
</head>
<body>
  <header>
    <h1>El tren de los heridos</h1>
  </header>
  <article>
    <p>Abre caminos de algodón profundo,<br/>
    amordaza las ruedas, los relojes,<br/>
    detén la voz del mar; de la paloma:<br/>
    emociona la noche de los sueños.<br/><br/></p>
  </article>
</body>
</html>
```

Poesía: El tren de los heridos

*Abre caminos de algodón profundo,
amordaza las ruedas, los relojes,
detén la voz del mar; de la paloma:
emociona la noche de los sueños.*

Miguel Hernández

1.5.3 SELECTORES PSEUDO-CLASES

Las pseudo-clases permiten determinar con una mayor exactitud el elemento sobre el que se quiere aplicar un estilo. CSS3 presenta grandes novedades incorporando numerosas formas de selección mediante el uso de pseudo-clases teniendo en cuenta el DOM.

En todas las pseudo-clases se puede indicar un elemento padre concreto o no, por ejemplo

- **p a: nth-child(1)**, selecciona el primer enlace de los elementos que están dentro de cualquier etiqueta <p> y a: nth-child(1), selecciona el primer enlace de cualquier padre.
A continuación se enumeran algunas de las pseudo-clases incluidas en CSS3.
- **elemento:nth-child (N)**, selecciona el elemento que es el N-ésimo hijo de su padre (el 1 es el que aparece primero en el código HTML y los siguientes se cuentan en orden secuencial dentro del mismo padre)
- **elemento:nth-last-child (N)**, es igual que el anterior pero empieza a contar por el último hijo, así el antepenúltimo será el número 2
- **elemento:first-child** y **elemento:last-child**, selecciona el primer y el último hijo del elemento indicado respectivamente.
- **elemento:nth-of-type(N)**, selecciona el elemento indicado con la restricción de que sea el N-ésimo elemento hermano de ese tipo (tipo elemento)
- **elemento:nth-last-of-type(N)**, igual que el anterior pero el elemento se empieza a contar desde el último hijo.

Si en lugar del número de elemento indicamos la palabra clave odd (impares) o even (par) se consigue hacer referencia a los elemento impares y pares respectivamente. Por ejemplo, p:nth-child(odd), selecciona todos lo párrafos impares.

1.5.4 OTROS SELECTORES

```

<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <style>
    p:first-child {background:#6F6}
    p:nth-child(2) {background:#ff0000}
    p:nth-last-of-type(4) {background:#C90}
    p:nth-of-type(4) {background:#39C}
    p:nth-last-child(2) {background:#CF0}
    p:last-child {background:#939}
    p:nth-child(odd) {font-style:italic}
  </style>
</head>
<body>
  <header>
    <h1>Gloria Fuertes</h1>
  </header>
  <article>
    <p>Doña Pito Piturra</p>
    <p>Tiene unos guantes,</p>
    <p>Doña Pito Piturra</p>
    <p>Muy elegantes.</p>
    <p>Doña Pito Piturra</p>
    <p>Tiene un sombrero,</p>
  </article>
</body>
</html>

```

Gloria Fuertes

Existen otros selectores CSS3 que aportan nuevas posibilidades entre los que se encuentran

- **elemento:not (elemento1)**, que selecciona todos los elementos de la página excepto los que no sean del tipo de **elemento1** indicado, **elemento1** puede ser una clase, un id, etc.

En la imagen siguiente podemos observar un ejemplo de este selector:

```

<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <style>
    a:not(.tipol) {font-style:italic}
    .tipol{color:red}
  </style>
</head>
<body>
  <nav>
    <a class="tipol" href="enlace1">Menu1</a>
    <a class="tipo2" href="enlace2">Menu2</a>
    <a class="tipol" href="enlace3">Menu3</a>
    <a class="tipo2" href="enlace4">Menu4</a>
  </nav>
</body>
</html>

```

2 COLOR

Las recomendaciones que el W3C elabora para las características de color de las hojas de estilo se publican en el documento llamado CSS Color Module Level 3 (Módulo de color CSS de nivel 3).

















Las propiedades de las hojas de estilo relacionadas con el color pueden afectar

- al color del texto,
- al color de los fondos de distintos contenedores (capas, celdas, etc.),
- al color de los bordes y
- al color de otras partes de un documento Web.

La recomendación CSS2 permite utilizar colores RRGGBB (formato hexadecimal), RGB (Red, Green, Blue) y nombres de colores (red, white, black, etc.).

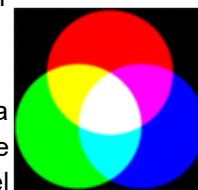
El módulo de color de CSS3 incrementa las codificaciones de colores y permite utilizar HSL (Hue, Saturation, Lightness), CMYK (Cyan, Magenta, Yellow, Black), HSLA (Hue, Saturation, Lightness, Alpha) y RGBA (Red, Green, Blue, Alpha). El código CMYK es el que lleva mayor retraso en su incorporación a los navegadores.

En la siguiente tabla podemos ver los colores básicos de la Web y sus códigos según el formato utilizado

Color	Nombre	Código RRGGBB	Código RGB	Código HSL	Código CMYK
	black	#000000	0,0,0	0,0%,0%	75,68,67,90
	silver	#C0C0C0	192,192,192	0,0%,75%	25,20,20,0
	gray	#808080	128,128,128	0,0%,50%	52,43,43,8
	white	#FFFFFF	255,255,255	0,0%,100%	0,0,0,0
	maroon	#800000	128,0,0	0,100%,50%	29,100,100,38
	red	#FF0000	255,0,0	0,100%,100%	0,99,100,0
	purple	#800080	128,0,128	300,100%,50%	61,100,14,3
	fuchsia	#FF00FF	255,0,255	300,100%,100%	27,82,0,0
	green	#008000	0,128,0	120,100%,50%	87,24,100,13
	lime	#00FF00	0,255,0	120,100%,100%	63,0,100,0
	olive	#808000	128,128,0	60,100%,50%	51,36,100,13
	yellow	#FFFF00	255,255,0	60,100%,100%	6,0,97,0
	navy	#000080	0,0,128	240,100%,50%	100,98,14,17
	blue	#0000FF	0,0,255	240,100%,100%	88,77,0,0
	teal	#008080	0,128,128	180,100%,50%	86,31,49,8
	aqua	#00FFFF	0,255,255	180,100%,100%	52,0,13,0

El formato RGB es uno de los más utilizados y obtiene los colores de la mezcla del rojo, verde y azul. Para hacer referencia a un color se debe indicar la proporción que éste contiene de rojo, de verde y de azul en unos valores que van del 0 al 255, así el amarillo se especifica como (255, 255, 0), lo que quiere decir que se obtiene de la mezcla de rojo y verde. Este formato puede sufrir grandes variaciones en el color resultante en función del dispositivo utilizado.

El formato RRGGBB se guía por los mismos conceptos que el RGB pero la notación utilizada es el código hexadecimal, donde los dos primeros dígitos se refiere al color rojo, los dos siguientes al color verde y el resto al color azul, así el color amarillo sería FFFF00 (el número 255 es FF en código hexadecimal).



El formato HSL responde a una representación 3D del color. HSL tiene en cuenta el matiz (rojo, azul, verde, etc.), la saturación y el brillo. La siguiente imagen muestra distintos niveles de saturación del color rojo.



El formato CMYK es un modelo de 32 bits que se basa en la obtención de colores mediante la mezcla del cian, magenta, amarillo y negro. Las imprentas offsets imprimen generalmente en estos cuatro colores. Este formato aún no está implementado en la mayoría de los navegadores.



CANAL ALPHA

CSS3 ha añadido una nueva característica de ajuste de color. Además del formato RGB y HSLA se puede utilizar RGBA y HSLA, donde el último parámetro indica el valor del canal alpha (opacidad/transparencia). El valor del canal alpha es un número entre 0 y 1, cuanto menor sea el número mayor grado de transparencia.

El hecho de utilizar códigos de color que admiten un valor para indicar el grado de transparencia/opacidad incrementa las posibilidades en el diseño Web y permite obtener efectos como los que se muestran a continuación con una sola imagen fuente:



Existen numerosas aplicaciones on-line para extraer un código de color distintos formatos, por ejemplo: Consulta las siguientes páginas

<http://www.villamalea.eu/aponline/colorar.php>

<http://www.pazosblogger.com/2010/01/generador-de-codigos-de-colores.html>

COLOR. EJEMPLOS

La propiedad de color se puede utilizar en múltiples propiedades por ejemplo, en el color de un elemento (color) y el color de fondo del mismo (background-color).



A continuación se muestran dos ejemplos de páginas Web donde se utilizan distintos formatos de color en una hoja de estilos incluida en la zona head de HTML

Ejemplo 1: El siguiente ejemplo utiliza como selectores etiquetas HTML y como códigos de color los nombres correspondientes.

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <style>
    body {color: blue; background: white }
    h1 {color: maroon }
    h2 {color: olive }
  </style>
</head>
<body>
  <header>
    <h1>Tienda on-line</h1>
    <h2>Compra con comodidad</h2>
  </header>
  <article>
    Compra con nosotros ropa, complementos y zapatos
  </article>
</body>
</html>
```

Ejemplo 2: El siguiente ejemplo utilizar selectores de atributo y pseudo-elementos para asignar color a distintos componentes del documento Web utilizando los códigos HSL, RGB y RGBA

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <title>Música</title>
  <style>
    /*Selecciona aquellos elementos cuyo id comienza por "Titulo"*/
    *[id^="Titulo"]{color:HSL(0, 100%, 25%)}
    /*Selecciona aquellos elementos que tienen class="Articulo"*/
    .Articulo{background-color:RGBA(192,192,192,0.3)}
    /*Selecciona la primera letra del bloque p que está inmediatamente después de h2*/
    h2+p::first-letter {font-size:30px;color:RGB(255,0,255);}
  </style>
</head>
<body>
  <header>
    <h1 id="Titulo1">TE VAKA</h1>
    <h2 id="Titulo2">MÚSICA DEL PACÍFICO</h2>
  </header>
```

```
<section id="Historia">
  <article>
    <h2 class="Articulo">ORIGEN</h2>
    <p>Originarios de Aotearoa (Nueva Zelanda)</p>
    <p>se identifican con la cultura maori.</p>
  </article>
  <article>
    <h2 class="Articulo">IDIOMA</h2>
    <p>El tokelaués es su lengua principal y en la que compone </p>
    <p>todas las canciones de Te Vaka.</p>
  </article>
</section>
</body>
</html>
```

Realiza pequeñas modificaciones sobre el las hojas de estilo de las páginas Web de los ejemplos para practicar con todos los formatos de color y comprender sus efectos. Utiliza los selectores de CSS3.

COLOR. TRANSPARENCIA

Hasta el momento, para manipular la transparencia de una imagen era necesario utilizar un programa de edición gráfica para modificar esta característica. Con el desarrollo de CSS3 se han creado distintos métodos para manipular la opacidad de una imagen o de un color asignado.

Para modificar la transparencia de un color ya hemos visto que se puede utilizar el canal alpha en los formatos RGBA y HSLA, pero también existe una propiedad específica denominada opacity que modifica la transparencia de cualquier propiedad de color o imagen.

El valor de la propiedad opacity se establece mediante un número decimal que se debe encontrar entre 0.0 y 1.0. Cuando menor sea el valor del atributo mayor será la transparencia aplicada. Así un valor 0.0 del atributo opacity implica que el objeto gráfico es invisible y si el valor es 1.0 será totalmente opaco, de esta forma, el valor 0.5 corresponde a un elemento semitransparente.

A continuación se puede observar el efecto de distintos grados de opacidad sobre una imagen.



0.9

Texto

0.7

Texto

0.5

Texto

0.3

Texto

0.2

Texto

Ejemplo 1: El siguiente código CSS3 fija distintas opacidades (0.3, 0.5 y 0.8) a las capas con id primero, segundo y tercero que tienen el mismo color de fondo (#00F).

Código CSS3

```
#primero, #segundo, #tercero {
    background-color:#00F;
    color:#FFF;          /*Color de texto*/
}
#primero {
    opacity:0.3;
}
#segundo {
    opacity:0.5;
}
#tercero {
    opacity:0.8;
}
```

Código HTML

```
<h1>Probando opacity</h1>
<div id="primero">
    La primera capa
</div>
<div id="segundo">
    La segunda capa
</div>
<div id="tercero">
    La tercera capa
</div>
```

Ejemplo 2: El ejemplo que se presenta a continuación muestra un estilo CSS3 aplicado a una imagen por medio de su id="tucan" que modifica la transparencia de la misma.

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset=utf-8">
<title>Imágenes con transparencia</title>
<style>
    #tucan {opacity:0.3;}
</style>
</head>
<body>
    <h1 >Probando opacity</h1>
    
</body>
</html>
```

3 UNIDADES DE MEDIDA

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



CSS divide las unidades de medida en dos grupos: absolutas y relativas. Las medidas relativas definen su valor en relación con otra medida, por lo que para obtener su valor real, se debe realizar alguna operación con el valor indicado.

Las unidades absolutas establecen de forma completa el valor de una medida, por lo que su valor real es directamente el valor indicado.

3.1.UNIDADES ABSOLUTAS

Una medida indicada mediante unidades absolutas está completamente definida, ya que su valor no depende de otro valor de referencia. A continuación se muestra la lista completa de unidades absolutas definidas por CSS y su significado:

- in, pulgadas
- cm, centímetros.
- mm, milímetros.
- pt, puntos.
- pc, picas.

3.2.UNIDADES RELATIVAS

Las unidades relativas, a diferencia de las absolutas, no están completamente definidas, ya que su valor siempre está referenciado respecto a otro valor. A pesar de su aparente dificultad, son las más utilizadas en el diseño web por la flexibilidad con la que se adaptan a los diferentes medios.

- em, relativa respecto del tamaño de letra del elemento. Hace referencia al tamaño en puntos de la letra que se está utilizando
- ex, relativa respecto de la altura de la letra x ("equis minúscula") del tipo y tamaño de letra del elemento.
- px, (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

3.3.PORCENTAJES

Un porcentaje está formado por un valor numérico seguido del símbolo % y siempre está referenciado a otra medida

3.4.RECOMENDACIONES

En general, se recomienda el uso de unidades relativas siempre que sea posible, ya que mejora la accesibilidad de la página y permite que los documentos se adapten fácilmente a cualquier medio y dispositivo.

4 EFECTOS CSS

4.1.SOMBRAS

Un efecto interesante que podemos utilizar en nuestros diseños web a partir de CSS3 es el de las **sombras**. A través de dos propiedades, `box-shadow` y `text-shadow`, se nos permite crear sombras a dos tipos de elementos diferentes respectivamente:

box-shadow: `<medida> <medida> <medida>? <medida>? <color>` → Aplica una sombra a una caja o un contenedor

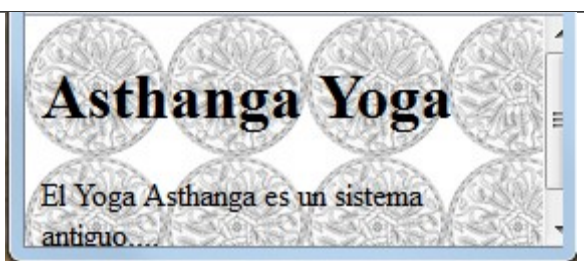
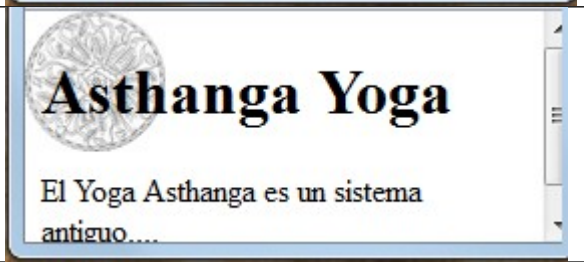
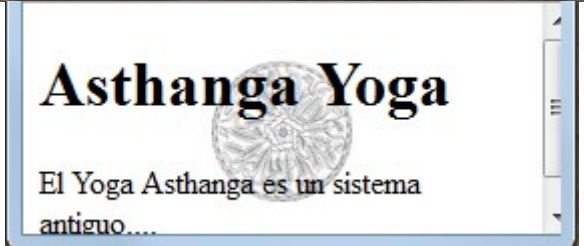
- La primera medida es obligatoria e indica el desplazamiento horizontal de la sombra. Si el valor es positivo, la sombra se desplaza hacia la derecha y si es negativo, se desplaza hacia la izquierda.

- La segunda medida también es obligatoria e indica el desplazamiento vertical de la sombra. Si el valor es positivo, la sombra se desplaza hacia abajo y si es negativo, se desplaza hacia arriba.
- La tercera medida es opcional e indica el radio utilizado para difuminar la sombra. Cuanto más grande sea su valor, más borrosa aparece la sombra. Si se utiliza el valor 0, la sombra se muestra como un color sólido.
- La cuarta medida también es opcional e indica el radio con el que se expande la sombra. Si se establece un valor positivo, la sombra se expande en todas direcciones. Si se utiliza un valor negativo, la sombra se comprime.

4.2.FONDOS

CSS3 aporta nuevas posibilidades para establecer fondos en una página Web o en una porción de la misma. En las recomendaciones de hojas de estilo anteriores sólo era posible introducir una imagen de fondo para cada contenedor, esta imagen podía mostrarse de forma única en distintas posiciones o bien de forma repetida. Cualquier necesidad de diseño que superase estas posibilidades era necesario afrontarse con trucos o utilizando un editor de imágenes.

A continuación se muestran algunas de las propiedades de los fondos en el nivel 2 de CSS

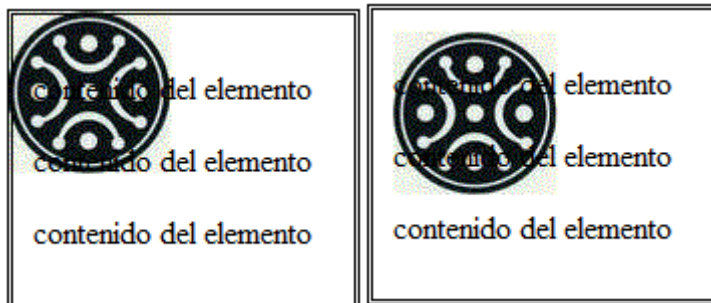
<p>Imagen de fondo que se muestra de forma repetida por todo el <i>body</i>.</p> <pre>body{ background:url(mandala.jpg) }</pre>	
<p>Imagen de fondo que se muestra de forma sólo una vez en la parte superior izquierda del <i>body</i>.</p> <pre>body{ background:url(mandala.jpg) no-repeat }</pre>	
<p>Imagen de fondo que se muestra de forma sólo una vez centrada en el <i>body</i>.</p> <pre>body{ background:url(mandala.jpg) no-repeat center }</pre>	

La nueva recomendación de nivel 3 de las hojas de estilo permite modificar las características de la imagen de fondo de un contenedor (página Web, div, iframe, etc.) dando la posibilidad de realizar las siguientes operaciones:

- Utilizar múltiples fondos para un mismo contenedor.



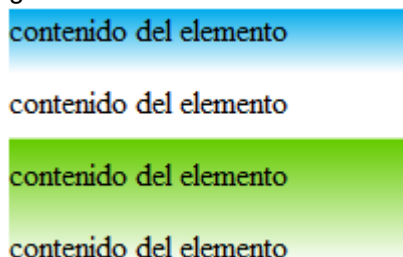
- Situar una imagen de fondo en distintas posiciones



- Cambiar el tamaño de la imagen para que se adapte al contenedor donde reside.



- Aplicar difuminados y gradientes.



A continuación analizaremos las nuevas propiedades de las hojas de estilo relacionadas con los fondos.

CSS3 permite añadir varios fondos. La técnica para realizar esta operación consiste en utilizar la propiedad **background** y fijar los atributos de cada una de las imágenes separadas por un espacio en blanco; los grupos de atributos de cada imagen se separarán con una coma.

Los atributos de uso más habitual para cada una de las imágenes son:

- url (localización del fichero de la imagen),
- posición (center, right, left, bottom, top) y
- repetición (no-repeat, repeat).

Se debe tener en cuenta que la imagen puesta en primer lugar es la que estará en primer plano en el caso de que se sobrepongan sobre alguna otra imagen.

Ejemplo

Código CSS

```
.fondos {
    background: url(simbolo1.gif) center no-repeat,
    url(simbolo2.gif) repeat;

    width: 300px;
}
```

url(simbolo2.gif) repeat;

Código HTML

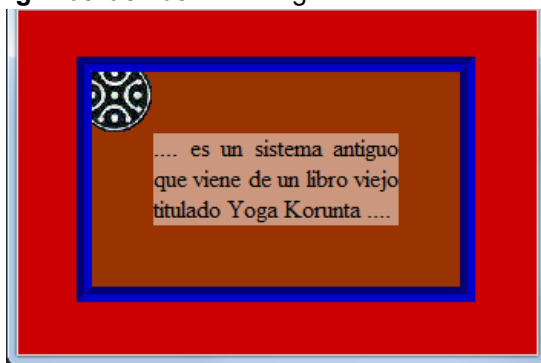
```
<div class="fondos">
    <p>contenido del elemento que va a tener 2 fondos </p>
    <p>contenido del elemento que va a tener 2 fondos </p>
    <p>contenido del elemento que va a tener 2 fondos </p>
    <p>contenido del elemento que va a tener 2 fondos </p>
    <p>contenido del elemento que va a tener 2 fondos </p>
    <p>contenido del elemento que va a tener 2 fondos </p>
</div>
```

Resultado en el navegador:

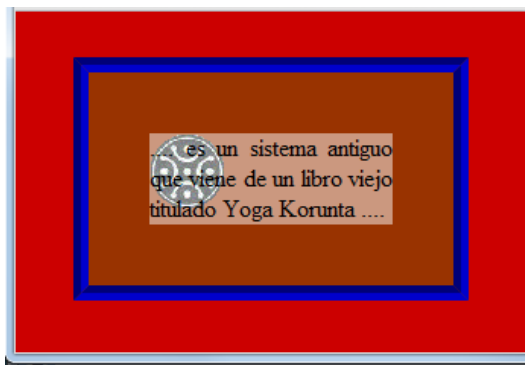


CSS3 utiliza la propiedad **background-origin** para indicar la posición de la imagen de fondo dentro de un elemento, con este atributo se pueden indicar tres posiciones relativas distintas para el extremo superior izquierdo de la imagen.

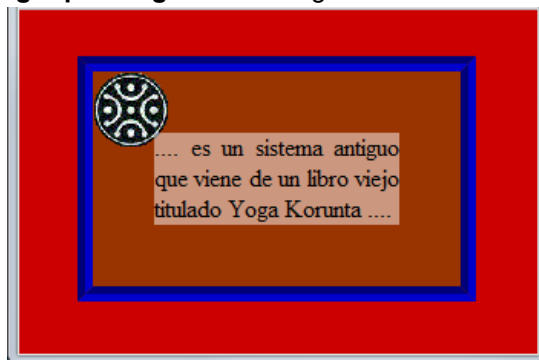
- **background-origin: border-box:** la imagen comenzará a colocarse en el borde.



- **background-origin: content-box:** la imagen comenzará a colocarse en la zona de contenido.

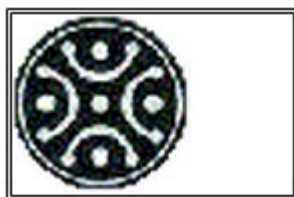


- **background-origin:padding-box:** la imagen comenzará a colocarse en la zona de padding.



El tamaño de una imagen utilizada como fondo puede llegar a adaptarse al tamaño de su contenedor utilizando la propiedad **background-size** y puede contener parámetros como los que se describen a continuación:

- **background-size: contain.** Indica que el fondo se debe adaptar a la altura o a la anchura del contenedor de tal forma que quepa toda la imagen, respetando las proporciones de la imagen.



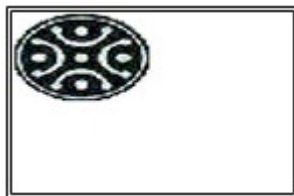
- **background-size: cover.** Indica que el fondo se debe escalar en función de las dimensiones del contenedor y como consecuencia puede no verse parte de la imagen.



- **background-size: valor1 px valor2 px.** En este caso se pueden especificar las dimensiones de la imagen de fondo en píxeles. Por ejemplo para background-size= 50px 50px; la imagen del ejemplo quedaría:



- **background-size: valor% valor2%.** En este caso se pueden especificar las dimensiones de la imagen de fondo en tanto por ciento respecto al tamaño del contenedor. Por ejemplo para background-size= 50% 50%; la imagen del ejemplo quedaría:



4.3. GRADIENTES

También llamado fondos degradados, es una técnica muy utilizada entre los diseñadores. Un degradado es un efecto gráfico que se realiza a partir de dos colores como mínimo, lo que muestra un degradado es el resultado de la evolución gradual de un color a otro.

Hay dos tipos de degradados los **lineales** y los **radiales**.

DEGRADADOS



lineal



radial

Los degradados lineales son aquellos cuya transición de un color a otro comienza en un borde y evoluciona al extremo opuesto.

Los degradados radiales tienen el origen de su transición en el centro y se desarrolla de forma radial.

Con anterioridad a la aparición de CSS3 la única posibilidad de utilizar este efecto en el diseño de las páginas Web era utilizando programas de edición gráfica. Ahora CSS3 aporta propiedades que permiten crear degradados desde la hoja de estilos.

Por ahora, no hay una forma estándar de establecer los gradientes y tanto los navegadores basados en el motor Webkit (Safari u Chrome) como Mozilla tienen sus propias propiedades.

Para el navegador Mozilla, se utiliza el prefijo -moz:

- -moz-linear-gradient([<punto> || <ángulo>]? <stop>, <stop> [, <stop>]*)
- -moz-radial-gradient([<posicion> || <ángulo>]? [<forma> || <tamaño>]? <stop>, <stop> [, <stop>]*)

Y para Webkit, es bastante diferente:

- -webkit-gradient(<tipo>, <punto> [, <radio>]?, <punto> [, <radio>]? [, <stop>]*)

Donde

- **tipo** sólo se utiliza en el gradiente de Webkit e indica si es lineal (linear) o radial (radial).
- **punto** se corresponde con un par de valores separados por espacios, puede ser indicado con números, porcentajes o las palabras clave "top", "bottom", "left" y "right".



- **radio** es un número y sólo se indica en el caso de que el tipo de degradado es radial.
- **sopt** se corresponde con una función con dos argumentos, un número y un color, indicando donde finaliza el gradiente, también admite el uso de las funciones from y to para indicar el principio y el fin de la transición.

Para crear propiedades CSS3 de degradados se pueden utilizar herramientas on-line como <http://westciv.com/tools/gradients/index.html> para los gradientes lineales, y para los gradientes radiales <http://westciv.com/tools/radialgradients/index.html>.

4.4.BORDES

Hasta ahora los bordes que permitían crear las hojas de estilo eran muy básicos, no era posible redondear las esquinas ni utilizar imágenes o degradados. Estos efectos sólo se lograban utilizando programas especializados en el tratamiento de imágenes o complejos scripts.

Con las hojas de estilo de nivel 3 las posibilidades gráficas se han incrementado exponencialmente y ya es factible realizar bordes utilizando exclusivamente propiedades de CSS3. Estas nuevas propiedades permiten redondear las esquinas, utilizar imágenes con marcos para rodear textos, rellenar el borde con un efecto de degradado y crear sombras.

Para la creación de propiedades que modifiquen los bordes podemos utilizar distintas herramientas on-line como las que se indican a continuación.

- Para crear bordes redondeados se puede hacer uso de la página Web <http://border-radius.com/>.
- Para desarrollar las propiedades que permiten crear bordes con sombras podemos utilizar la herramienta on-line que se encuentra en <http://css3generator.com/>. También esta herramienta permite crear bordes redondeados utilizando la opción Border Radius de la lista desplegable.

La propiedad border-radius que permite definir el radio de las curvas de las esquinas aún no ha sido incorporada por los navegadores y tanto Mozilla como los navegadores basados enWebKit (Chrome y Safari) han implementado una propiedad a partir de ésta.

Mozilla → -moz-border-radius

WebKit → -webkit-border-radius

Si se asigna un valor (en píxeles) a la propiedad, entonces el radio aplicado es el mismo para todas las esquinas. Pero se le pueden asignar hasta 4 valores, uno para cada esquina (orden: sup-izquierda, sup-derecha, inf-derecha, inf-izquierda). Para que los bordes se redondeen la propiedad border debe tener el atributo solid.

Ejemplo: El siguiente estilo muestra las propiedades para hacer un borde redondeando sus esquinas de manera uniforme.

```
div {
    border: 1px solid #000000;
    -moz-border-radius: 7px;
    -webkit-border-radius: 7px;
    padding: 10px;}
```

Se pueden especificar valores diferentes para cada ángulo del borde, como podemos ver a continuación:

BORDES CON IMÁGENES

La propiedad que permite introducir imágenes en los bordes es **border-image**. Esta propiedad permitirá utilizar una o varias imágenes como bordes.

Esta propiedad aún está ausente en muchos navegadores y por ahora sólo podemos contar con Firefox y los navegadores basados en el motor enWebKit (Chrome y Safari) para ver este efecto.

La sintaxis de la propiedad que permite utilizar una imagen como borde es la siguiente:

`border-image: url(ficheroImagen) montajImagen transformacionImagen`

Donde

- **ficheroImagen.** Es la ruta de la imagen que se va a utilizar como borde, se pueden manejar hasta tres imágenes.
- **montajImagen.** Son las propiedades que se utilizan para especificar el tamaño que se va a utilizar de cada esquina de la imagen (en el orden de las agujas de reloj). Se pueden expresar en porcentajes y, en el caso de utilizar un valor sin unidad de medida se consideran que son píxeles. Si sólo se pone un valor se considera que este afecta a todos los bordes, si se ponen dos entonces el primero es para los lados superior e inferior y el segundo para los lados izquierdo y derecho.
- **transformacionImagen.** Son las propiedades que se utilizan para determinar como se van a escalar y cortar los bordes. Puede utilizar tres posibles valores: stretch (estirar), repeat (repetir) y round (redondear). En el caso de especificar un solo valor se considera que se aplica sobre todos los lados del borde.

Ejemplo:

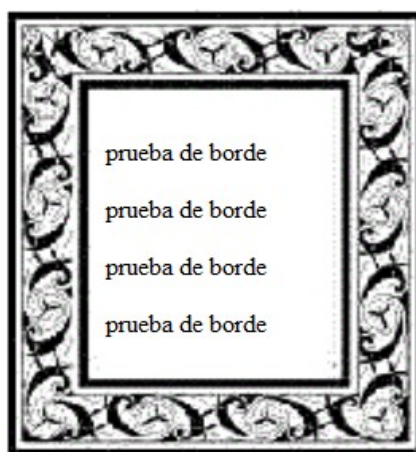
Para comprender el efecto de estas propiedades, podemos observar el efecto del código siguiente

Código HTML

```
<div class="borde">
  <p>prueba de borde</p>
  <p>prueba de borde</p>
  <p>prueba de borde</p>
  <p>prueba de borde</p>
</div>
```

Código CSS

```
.borde { width: 120px; height: 150px; padding: 40px;
  border: 2px;
  -webkit-border-image: url(marco1.gif) 5% stretch;
  -moz-border-image: url(marco1.gif) 5% stretch;}
```



Puedes utilizar el siguiente enlace para ampliar la información del uso de las imágenes como bordes: <http://www.w3.org/TR/css3-background/#border-images>

4.5.TEXTO.



Hasta la aparición de las hojas de estilo CSS3, una de las mayores limitaciones para los diseñadores Web era la imposibilidad de controlar la tipografía con que se mostraban los textos en las páginas Web.

Cuando un navegador muestra contenidos de tipo texto de una página Web busca las fuentes (tipos de letra) en el ordenador cliente. Debido a este funcionamiento, sólo se pueden utilizar aquellas fuentes que habitualmente se encuentran en los equipos de los usuarios.

En caso de que una página Web demande el uso de una tipografía que no existe en el equipo cliente, el navegador la substituye por la fuente considerada por defecto, alterando de esta manera el diseño establecido para el documento.

La consecuencia directa de esta limitación es que todos los diseñadores se limitan a utilizar aquellas fuentes que casi todos los usuarios tienen instaladas en sus equipos que suelen ser Arial, Verdana, Tahoma, Courier, Times New Roman o Georgia.

En el estándar CSS2 se definió una propiedad `@font-face` que permitía describir las fuentes que utiliza una página Web, pero esta propiedad se anuló en la versión CSS2.1. CSS3 ha vuelto a recuperar esta directiva en el módulo llamado Web Fonts dejando vía libre al uso de cualquier fuente siempre que ésta esté instalada en el servidor.

El formato de la directiva `@font-face` es la siguiente:

```
@font-face{
    font-family: nombre_fuente;
    src:source;
    [font-weight: peso];
    [font-style: estilo];
}
```

Donde

- nombre fuente es el nombre que se utilizará como alias de la tipografía
- source es la ruta del servidor donde se encuentra el archivo de la tipografía, puede haber más de una ruta separada por comas
- peso es el grado de oscuridad y el grosor de la fuente de la fuente (esta propiedad es opcional), puede tener los siguientes valores: `normal` | `bold` | `bolder` | `lighter`
- estilo es el estilo que se le debe aplicar al tipo de letra (esta propiedad es opcional), puede contener los siguientes valores: `normal` | `italic` | `oblique`

Se debe crear una directiva `@font-face` para cada fuente que se desee descargar del servidor para utilizar en un documento Web.

La propiedad `font-weight` determina el grosor de la fuente sus valores habituales son: **normal** | **bold** | **bolder** | **lighter**.

La propiedad `font-style` permite modificar la fuente para que el texto se muestre en cursiva u oblicua, sus valores habituales pueden ser: **normal** | **italic** | **oblicua**.

Para utilizar la fuente definida con `@font-face` tan sólo es necesario utilizar la propiedad `font-family` para referenciarla.

```
selector {
    font-family: nombre_fuente;
}
```

Se puede modificar el tamaño de la fuente utilizando la propiedad CSS `font-size`, esta propiedad permite utilizar las siguientes unidades:



Unidades absolutas	Unidades relativas
<p>pulgadas (in). <i>Una pulgada=2.54 cm.</i></p> <p>centímetros (cm).</p> <p>milímetros (mm).</p> <p>puntos (pt). <i>Un punto=1/72 de pulgada.</i></p> <p>picas (pc). <i>Una pica=12 puntos</i></p>	<p>em. <i>Es relativo al tamaño de la M de la fuente instalada.</i></p> <p>ex. <i>Es relativo a la altura de la x minúscula de la fuente utilizada.</i></p> <p>px. <i>Es un píxel y es relativo a la resolución de la pantalla.</i></p>

Ejemplo: La siguiente página Web muestra un texto donde se aplican estilos CSS3 con fuentes personalizadas: Cake.ttf y East-Side-Italic.ttf.

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <title>Documento sin título</title>
```

```
  <style>
```

```
    @font-face {
```

```
      font-family: Titulo;
```

```
      src: url(Cake.ttf);
```

```
      font-weight:bold;}

```

```
    @font-face {
```

```
      font-family: Parrafo;
```

```
      src: url(East-Side-Italic.ttf);}

```

```
  h1 { font-family: Titulo;}

```

```
  div {font-family: Parrafo;}

```

```
</style>
```

```
</head>
```

```
<body>
```

```
  <h1>Probando fuentes</h1>
```

```
  <div>
```

CSS3 describe cómo se especifican las propiedades de fuente y la cantidad de recursos de fuentes que se cargan de forma dinámica....

```
  </div>
```

```
</body>
```

```
</html>
```

Probando fuentes

CSS3 describe cómo se especifican las propiedades de fuente y la cantidad de recursos de fuentes que se cargan de forma dinámica....

Se pueden descargar fuentes gratuitas en la red en múltiples direcciones Web, por ejemplo:

<http://www.fontreactor.com/>

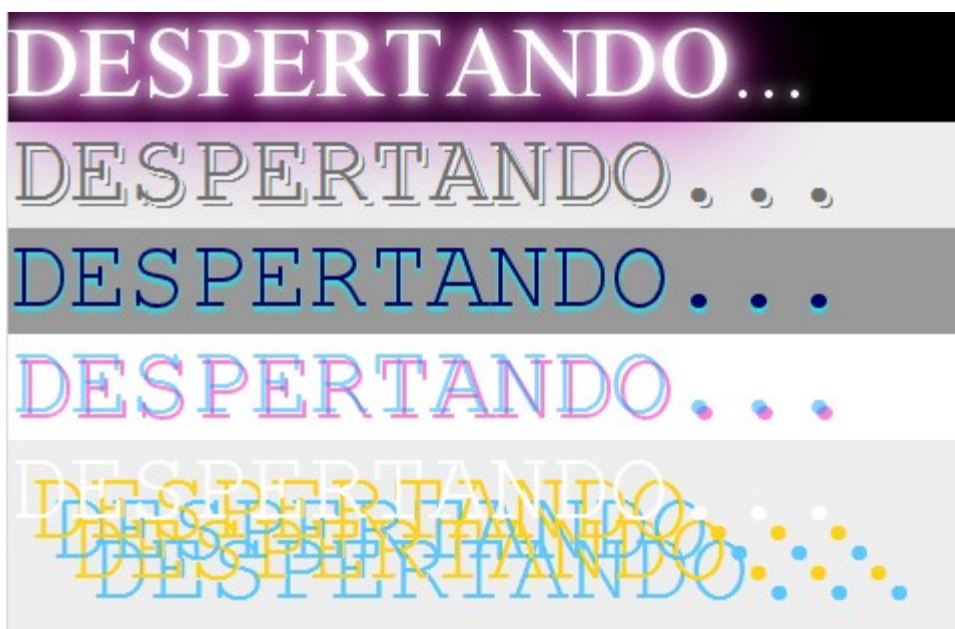
<http://www.letramania.com>

4.5.1 Efectos de texto

Con las propiedades de CSS3 es posible realizar múltiples efectos sobre el texto. El nuevo estándar de las hojas de estilo permite añadir sombras al texto, crear contornos, rotar el texto, etc. Hasta ahora este tipo de características sólo era posible realizarlas utilizando editores gráficos como PhotoShop.

En general, las propiedades que permiten modificar el texto para añadirle sombras, contornos y hacer rotaciones no están agregadas a los motores de los navegadores en su notación estándar sugerida por el W3C.

La siguiente imagen muestra distintos efectos logrados añadiendo sombras al texto.



El efecto contorno proporciona propiedades para crear contornos en el texto como podemos ver en la siguiente imagen

Caminando...

Las nuevas propiedades que se están desarrollando en el documento CSS Text Level son:

- Para introducir efectos de sombreado al texto: **text-shadow**
- Para crear efectos de contorno en un texto: **text-stroke-color** y **text-stroke-width**

Para crear las propiedades CSS3 que provocan distintos efectos sobre el texto se puede acudir a la herramienta on-line <http://www.css3maker.com>

Para construir estilos que realizan sorprendentes transformaciones en el texto se recomienda consultar la siguiente herramienta online <http://csswarp.eleqtriq.com/>

La propiedad **text-shadow** es una herramienta CSS3 útil y potente que permite realizar diseños atractivos añadiendo sombras al texto. Las sombras pueden configurarse seleccionando cualquier color, distancia y desenfoque.

La propiedad **text-shadow** tiene la siguiente sintaxis:

```
text-shadow: desp_h desp_v desenfoque color
```

Donde

- **desp_h, desp_v:** estos datos corresponden respectivamente con el desplazamiento horizontal y vertical de la sombra con respecto al texto; admite las distintas unidades de medida excepto el porcentaje.
- **desenfoque:** este parámetro indica el grado de desenfoque de la sombra; al utilizar un valor mayor que 0 para este parámetro la sombra se muestra con los bordes difuminados.
- **color:** es el color de la sombra, este parámetro se puede utilizar en primer lugar o en último lugar sin alterar en ningún caso el funcionamiento de la propiedad.

En una misma propiedad **text-shadow** se puede introducir los datos de más de una sombra, tan sólo es necesario separar una de otra con una coma

```
text-shadow: desp_h1 desp_v1 desenfoque1, color1, desp_h2 desp_v2 desenfoque2  
color2
```

Ejemplo: El siguiente estilo crea un efecto de una sombra sobre un mismo texto. En texto está en color #003 (azul oscuro), la sombra tiene color #CF0 (amarillo)

CSS

```
#sombra {  
margin: 200px auto;  
font-family: "League-Gothic", Courier;  
font-size: 40px; text-transform: uppercase;  
color: #003;  
text-shadow: 2px 2px 0px #CF0;}
```

HTML

```
<body>  
<article id="sombra">  
    En un lugar de la Mancha ...  
</article>  
</body>
```

EN UN LUGAR DE LA MANCHA ...

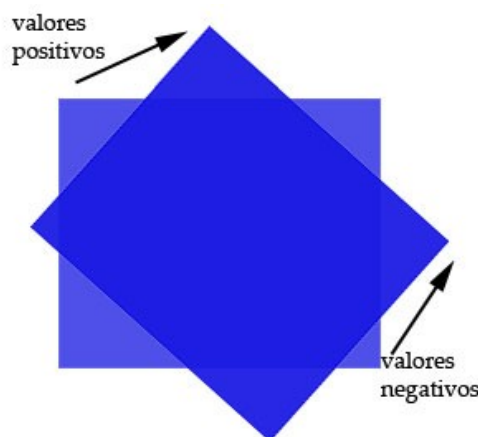
4.6. TRANSFORMACIONES

CSS3 realiza una gran aportación al dinamismo de las páginas Web mediante el uso de la propiedad **transform**, con esta propiedad se pueden rotar, inclinar o escalar cualquier elemento de DOM.

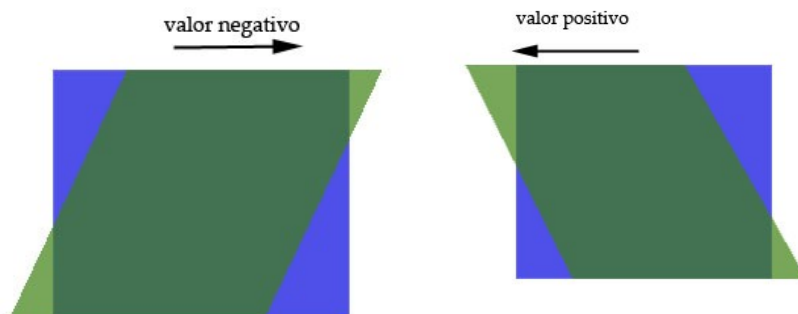
La sintaxis de la propiedad **transform** es la siguiente: **transform: tipo(cantidad)**

- **None.** Especifica que ninguna transformación puede ser aplicada

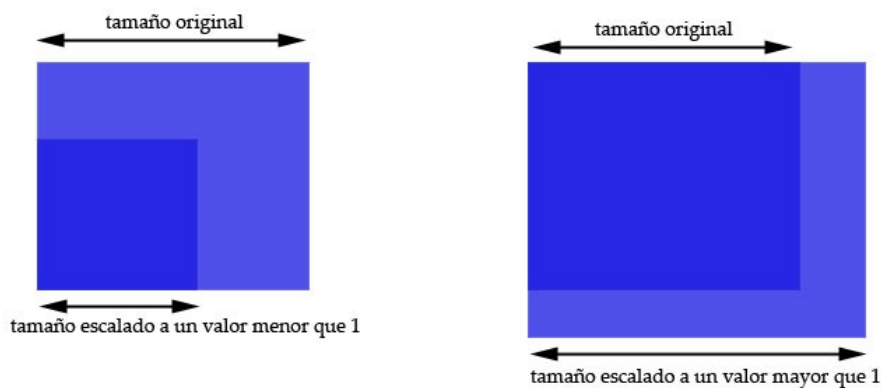
- **Rotación (tipo=rotate).** Permite rotar elementos provocando transformaciones como las que se muestran a continuación.



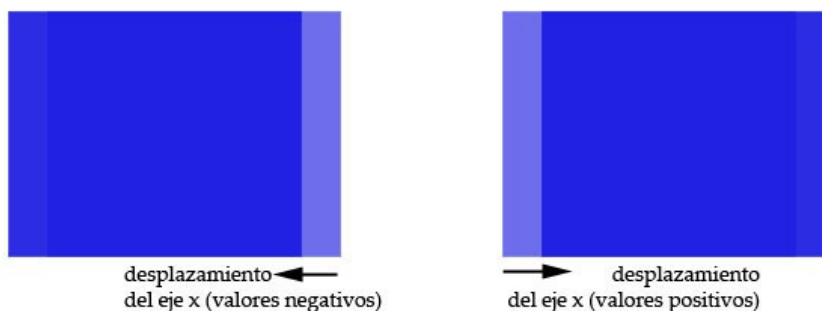
- **Inclinar (tipo=skew).** Con este tipo de transformación es posible desplazar los ejes horizontal y vertical del elemento en el que se aplica produciendo efectos de inclinación. En la actualidad no debe usarse. En este momento se utiliza la propiedad **skewX** → Inclinar un elemento a lo largo del eje X por el ángulo derecho, y la propiedad **skewY**



- **Cambiar el tamaño (tipo=scale).** Se puede transformar un elemento a un tamaño menor o mayor utilizando este tipo.



- **Mover el objeto (tipo=translate).** Este atributo nos da la posibilidad de realizar desplazamientos con los objetos. En la actualidad se utiliza **translateX**, **translateY**, **translate3d**



4.7. TRANSICIONES Y ANIMACIONES

Otra sorprendente característica de CSS3 es la posibilidad de realizar animaciones sin necesidad de recurrir a tecnologías extras (Flash, Silverlight, JavaScript, etc), tan sólo con el uso de hojas de estilo y HTML es posible crear llamativos efectos animados.

Una de las ventajas de utilizar CSS3 para la creación de animaciones es que nuestra Web puede ser más accesible, pues el uso de los plugins que necesitan Flash o Silverlight impide que la página sea universal. Utilizando CSS3 también evitamos la programación de complejos scripts y el uso de librerías como JQuery. En conclusión, simplificamos las herramientas que se necesitan para la creación de efectos animados y eso implica que se el funcionamiento y desarrollo de la Web se optimiza.

Para crear una animación con CSS3 es necesario determinar como mínimo dos estados, un estado inicial y un estado final. Cada uno de los estados se caracteriza por un conjunto determinado de propiedades y la evolución de uno a otro conforma el efecto de la animación.

Existen dos tipo de animaciones que se pueden crear utilizando CSS3

- Transiciones.
- Animaciones.

La transición es el tipo más simple de animación, se consigue al provocar un cambio de un estilo inicial a un estilo final en un tiempo determinado. La transición entre un estado y otro se indica por medio de propiedades que definen el tiempo invertido en el cambio de un estilo a otro, el retraso en el comienzo del efecto y la variación en la velocidad de la animación.

Todas las características de las transiciones se encuentran en el módulo Transitions en la dirección <http://www.w3.org/TR/css3-transitions>.

Las animaciones son más complejas y pueden variar las propiedades de un elemento tantas veces como se indique. Esta operación se realiza fijando fotogramas clave (keyframes). En cada keyframe se incluyen un grupo de propiedades que son las que el navegador cambiará en un tiempo definido para un objeto determinado (capa, imagen, etc.)

Para conocer más de las características de las animaciones se puede consultar el documento publicado por el W3C <http://www.w3.org/TR/css3-animations/>

4.7.1 TRANSICIONES

Uno de los efectos más utilizados hasta ahora era la pseudoclase :hover, la cual permite aplicar diferentes estilos en dos estados diferentes. Las transiciones permiten aplicar estilos diferentes entre estados de una manera más amplia.

Una de las primeras ventajas de las transiciones es que se puede utilizar una medida de tiempo a través de la cual se aplica el cambio de estilo.

- **transistion-property:** Indica a que atributos aplicamos la transición;
 - **none:** no hay ninguna propiedad afectada por la transición

- **all:** la transición se efectuará sobre todas las propiedades
- **nombrePropiedad:** esta es la propiedad sobre la que se efectuará la transición.
- **transition-duration:** Tiempo de duración de la transición; Se pueden utilizar como unidades los segundos (s) o los milisegundos (ms)
- **transition-timing-function:** La forma de ir cambiando con las opciones *ease*, *linear*, *ease-in*, *ease-out*, *ease-in-out*, *cubic-bezier*;

- *ease* - especifica un efecto de transición con un inicio lento, luego rápido, luego finaliza lentamente (esto es predeterminado)
- *linear* - especifica un efecto de transición con la misma velocidad de principio a fin
- *ease-in* - especifica un efecto de transición con un inicio lento
- *ease-out* - especifica un efecto de transición con un final lento
- *ease-in-out* - especifica un efecto de transición con inicio y fin lentos
- *cubic-bezier(n,n,n,n)* - le permite definir sus propios valores en una función cubic-bezier

- **transition-delay:** puede ponerse un valor de retardo, incluso negativo, para que la transición se aplique con un efecto de que ya ha progresado.
- **transition:** se puede utilizar esta propiedad para introducir todas las propiedades anteriores en una sola, utilizando la siguiente sintaxis:
 - **transition:** transition-property transition-duration transition-delay transition-timing-function

4.7.2 ANIMACIONES

Se pueden realizar animaciones sencillas utilizando CSS3 de forma exclusiva, para ello es necesario crear los distintos **keyframes**. Cada *keyframe* es un hilo de la animación por el que tiene que transcurrir el movimiento. Cualquier animación consta de dos *keyframe* como mínimo: el punto inicial y el punto final.

En primer lugar debemos definir la animación y sus keyframes, para ello utilizamos la directiva **@keyframes** en la hoja de estilos. Podemos definir tantas animaciones como se deseen utilizar en la página Web

La directiva **@keyframes** se define según la siguiente sintaxis.

```
@keyframes nombreAnimación{
    nombreKeyframe1{propiedades CSS}
    nombreKeyframe2{ propiedades CSS }
    .....
    nombreKeyframeN{ propiedades CSS }
}
```

Donde

keyframes es la palabra reservada para la definición de la animación

nombreAnimación es el nombre que va a identificar la animación

nombreKeyframeNº puede ser *from*, *to*: si la animación consta sólo de dos keyframes se indica el inicial con *from* y el final con *to*.

- **nº %:** si la animación consta de más de dos keyframes se debe indicar el tanto por ciento del tiempo del total de la animación en el que se debe ejecutar cada keyframe el número inicial debe ser el 0, el final el 100 y los del medio cualquiera entre 0 y 100.

A continuación se muestran dos ejemplos tipo de animaciones donde se utilizan dos y varios keyframes.

La siguiente animación provoca un ensanchamiento de una capa en dos pasos.

```
@keyframes MiAnimacion {
    from {width:100px;}
```

```

        to {width: 200px;}
    }
    2. la siguiente animación provoca un ensanchamiento de una capa en 5 pasos.
    @keyframes MiAnimacion {
        0% {width:100px;}
        20% {width:120px;}
        60% {width:120px;}
        80% {width:120px;}
        100% {width: 200px;}
    }

```

Para poder ver en acción una animación definida con @keyframes es necesario relacionarla con un objeto HTML por medio de un selector, para ello debemos utilizar las siguiente propiedades:

- animation-name: contiene el nombre de la animación especificada en un @keyframes.
- animation-duration: determina la duración de la animación en segundos (s) o milisegundo (ms).

También podemos utilizar otras propiedades para aportar más parámetros al funcionamiento de la animación:

- animation-iteration-count: contiene un número entero que indica el número de veces que se va a repetir la animación.
- animation-timing-function: en esta propiedad podemos definir cómo progresa la animación entre los keyframes mediante ecuaciones matemáticas.
- animation-direction: si asignamos el valor `altérnate` a esta propiedad, la animación se producirá también en sentido inverso, para este caso, se tendrá en cuenta la animación de regreso al primer fotograma en el número de iteraciones establecidas en la propiedad `animation-timing-function`.
- animation-delay: determina el retraso del comienzo de la animación en segundos (s) o milisegundo (ms).
- animation: esta propiedad combina las anteriores de una forma resumida.

A la hora de aplicar una animación podemos optar por dos métodos que se describen a continuación:

- **A un selector de elemento HTML.** En este caso, la animación se ejecutará al iniciar la página Web.

En el siguiente ejemplo se aplica la animación a un div y ésta se ejecuta al cargar la página

```

div{
    width:100px;
    background:#0F3;
    animation-name: MiAnimacion;
    animation-duration: 10s;
    animation-iteration-count: 2;
    animation-direction: altérnate;
}

```

- **A una pseudoclase.** En este caso, la animación se ejecutará cuando se produzca el evento indicado en la pseudoclase.

En el siguiente ejemplo se aplica la animación a la pseudoclase `div:hover` y ésta se ejecuta cuando el ratón pasa por encima de la capa.

```

div:hover{
    animation-name: MiAnimacion;
    animation-duration: 10s;
    animation-iteration-count: 2;
}

```


animation-direction: alternate;

}