

Sumario

1 INTRODUCCIÓN AL CSS.....	2
2 VARIABLES.....	4
2.1 CONCEPTOS DE MARGEN, PADDING Y BORDE.....	5
3 SELECTORES.....	6
3.1 Selectores Simples.....	6
3.2 Selectores Combinados.....	6
3.3 Selectores de Atributos.....	7
3.4 Selectores Pseudo-clases.....	8
3.5 Selectores Pseudo-elementos.....	8
3.6 Otros selectores.....	10
4 Color.....	11
5 Gradientes.....	16
6 Bordes.....	18
6.1 Bordes con imágenes.....	19
7 Texto.....	20
7.1 Efectos de texto.....	22
CSS.....	24
HTML.....	24

INTRODUCCIÓN AL CSS – PARTE 1

1 INTRODUCCIÓN AL CSS

CSS3 es el lenguaje de Hojas de Estilo en Cascada (por sus siglas en inglés, Cascading Style Sheets) que se usa para definir la presentación visual y el diseño de documentos escritos en un lenguaje de marcado, como HTML.

Su función principal es separar la estructura del contenido de su apariencia visual, permitiendo controlar fuentes, colores, bordes, diseño, y agregar efectos como gradientes y animaciones para hacer las páginas web atractivas y funcionales en diversos dispositivos

CSS significa hojas de estilo en cascada (cascading style sheets), y es muy importante entender la palabra cascada. La forma en que se comporta la cascada es la clave para comprender el CSS.

En algún momento trabajarás en un proyecto y encontrarás que el CSS que pensabas que debería aplicarse a un elemento no funciona. Por lo general, el problema suele ser que has creado dos normas que podrían aplicarse al mismo elemento. La cascada, y el concepto estrechamente relacionado de especificidad son mecanismos que controlan qué regla se aplica cuando aparecen tales conflictos. Es posible que la regla que se aplique finalmente a tu elemento no sea la que esperas, por lo que debes comprender cómo funcionan estos mecanismos.

También es significativo el concepto de herencia, que significa que algunas propiedades CSS heredan por defecto los valores establecidos en el elemento padre, pero otras no. Esto también puede causar una respuesta diferente a la que esperas.

Cascada

En un primer nivel de simplicidad, la cascada en las hojas de estilo significa que el orden de las reglas importa en CSS: cuando dos reglas tienen la misma especificidad, se aplica la que aparece en último lugar en el CSS.

Especificidad

La especificidad es el modo que tiene el navegador de decidir qué regla se aplica si diversas reglas tienen selectores diferentes pero podrían aplicarse a un mismo elemento. Básicamente, la especificidad mide cuán específica es la selección de un selector:

- Un selector de elemento es menos específico (selecciona todos los elementos de aquel tipo que aparecen en la página) por lo que presenta una puntuación más baja en especificidad.
- Un selector de clase es más específico (selecciona solo los elementos de una página que tienen un valor de atributo class dado), y por tanto recibe una puntuación mayor.

Veamos un ejemplo. Aquí abajo encontrarás dos reglas que pueden aplicarse al elemento h1.

```
.main-heading {  
    color: red;  
}
```

```
h1 {  
    color: blue;  
}
```

```
<h1 class="main-heading">This is my heading.</h1>
```

Este elemento h1 termina siendo de color rojo: el selector de clase confiere a esta regla una mayor especificidad, así que se aplicará a pesar de la regla para el selector de elemento que aparece más abajo en el orden del código

Herencia

La herencia también debe entenderse en este contexto: algunos valores de las propiedades CSS que se han establecido para los elementos padre los heredan los elementos hijo, pero otros no.

Por ejemplo, si para un elemento se establece el color (color) y el tipo de letra (font-family), cada elemento que se encuentre dentro de él también se mostrará de ese color y con ese tipo de letra, a menos que les se haya aplicado un color y un tipo de letra diferentes directamente

Pautas a seguir

El orden en que se aplican las reglas CSS se determina por la especificidad y la cascada.

- Estilos en línea: Son los más específicos y tienen la mayor prioridad.
- ID (`<code>#id</code>`): Tienen la segunda mayor prioridad.
- Clases (`<code>.clase</code>`), Pseudoclases (`<code>:hover</code>`) y Atributos (`<code>[type="text"]</code>`): Tienen una prioridad menor que los ID.

- Tipos de elemento (<code>p</code>, <code>div</code>): Son los menos específicos.

Mejores Prácticas de Organización de Selectores

Agrupar selectores: Si varias reglas comparten las mismas propiedades, puedes agrupar los selectores para hacer el código más conciso y fácil de leer.

Código

```
h1, h2, h3 {  
  color: var(--color-primario);  
  margin-bottom: 1em;  
}
```

- Usa un orden lógico:** Organiza tus selectores y propiedades de forma lógica, como por componente o por sección del documento.
- Evitar la especificidad excesiva:** Intente usar la menor especificidad posible (clases sobre ID) para facilitar el mantenimiento y evitar conflictos.
- Declaraciones de Propiedades:** Cada declaración CSS consiste en una propiedad y un valor, separados por dos puntos (:) y seguidos por un punto y coma.
propiedad: valor;

Las reglas CSS son un conjunto de una o más declaraciones para uno o más selectores.

2 VARIABLES

Para un código CSS organizado, declara las variables de manera global en :root, utiliza el orden de especificidad de CSS (estilos en línea > ID > clases/pseudoclases/atributos), y agrupa selectores para una mejor legibilidad.

Declaración de Variables (Propiedades Personalizadas)

- Declaración:** Usa la sintaxis --nombre-variable: valor; dentro de un selector.
- Globalización:** Declara las variables en el selector :root para que estén disponibles en todo el documento.

Código

```
:root {  
  --color-primario: #3498db;  
  --tamaño-fuente: 16px;  
}
```

- **Uso:** Acceda a la variable usando la función var() con el nombre de la variable entre paréntesis.

Ejemplo

```
body {
  color: var(--color-primario);
  font-size: var(--tamanio-fuente);
}
```

2.1 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).

3 SELECTORES

Un selector CSS es una referencia a uno o varios elementos HTML, escrito generalmente desde código CSS (también pueden utilizarse desde ciertos lugares en Javascript). Es una forma rápida y cómoda de hacer referencia a elementos individuales o grupos de elementos, para posteriormente darles estilo.

Cuando estamos en una fase de aprendizaje muy inicial con CSS, es habitual dar estilo con selectores muy sencillos, donde generalmente utilizamos un selector genérico que representa una etiqueta HTML (por ejemplo div). Sin embargo, con el tiempo aprendemos a utilizar selectores más complejos que nos permitirán hacer más cosas y escribir menos.

La verdadera potencia de CSS radica en la capacidad de poder seleccionar sólo los elementos que nos interesen, ya que a medida que el documento HTML crezca, aparecerán nuevos elementos que podrían adoptar ese estilo, y eso es algo que, probablemente, no nos va a interesar.

Aquí tienes un resumen de los principales tipos de selectores en CSS3:

3.1 Selectores Simples

- **Selector de tipo:** Selecciona elementos por su nombre de etiqueta.
`p { color: blue; } /* Aplica a todos los párrafos */`
- **Selector de clase:** Selecciona elementos con una clase específica.
`.miClase { font-size: 16px; }`
- **Selector de ID:** Selecciona un elemento con un ID único.
`#miID { background-color: yellow; }`
- **Selector universal:** Selecciona todos los elementos.
`* { margin: 0; padding: 0; }`

3.2 Selectores Combinados

- **Selector descendente:** Selecciona elementos dentro de un contenedor.

`div p { color: red; } /* Todos los <p> dentro de un <div> */`

- **Selector hijo directo:** Selecciona solo los hijos directos.
`div > p { color: green; }`
- **Selector adyacente:** Selecciona el elemento inmediatamente siguiente.
`h1 + p { font-weight: bold; }`
- **Selector de hermanos generales:** Selecciona todos los hermanos siguientes.
`h1 ~ p { color: gray; }`

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

3.3 Selectores de Atributos

- **Por atributo exacto:**

`input[type="text"] { border: 1px solid black; }`

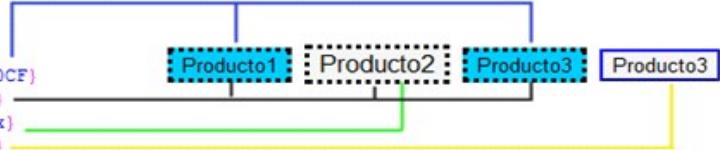
- **Por atributo que contiene una palabra específica:**

`[title~= "ejemplo"] { color: purple; }`

- **Por atributo que empieza con un valor:**
`[href^="https"] { text-decoration: underline; }`
- **Por atributo que termina con un valor:**
`[src$=".jpg"] { border: 2px solid red; }`
- **Por atributo que contiene una cadena específica:**
`[class*="boton"] { background-color: orange; }`

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>
```



3.4 Selectores Pseudo-clases

- **Estado de elementos:**

`a:hover { color: green; } /* Cuando el cursor pasa sobre un enlace */`
`input:focus { border-color: blue; }`

- **Posición de elementos:**

`li:first-child { font-weight: bold; }`
`li:last-child { font-style: italic; }`
`li:nth-child(2) { color: red; }`

3.5 Selectores Pseudo-elementos

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**

Ejemplo: Añadir un “borde interior” o una sombra sin afectar la estructura principal.

```
.cuadrado::before {  
    content: " ";  
    position: absolute;  
    top: 10px;  
    left: 10px;  
    right: 10px;  
    bottom: 10px;  
    border: 2px dashed #fff;  
}
```

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>
        h1::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

*A*bre 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

- **Estilizar partes específicas de un elemento:**

```
p::first-line { font-size: 18px; }
p::first-letter { color: red; }
```

3.6 Otros selectores

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(.tipo1){font-style:italic}
        .tipo1{color:red}
    </style>
</head>
<body>
    <nav>
        <a class="tipo1" href="enlace1">Menu1</a>
        <a class="tipo2" href="enlace2">Menu2</a>
        <a class="tipo1" href="enlace3">Menu3</a>
        <a class="tipo2" href="enlace4">Menu4</a>
    </nav>
</body>
</html>
```



4 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
- color de los fondos de distintos contenedores (capas, celdas, etc.),
- color de los bordes y
- color de otras partes de un documento Web.

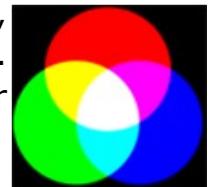
La recomendación CSS2 permite utilizar colores **RRGGBB** (formato hexadecimal), **RGB** (Red, Green, Blue) y nombres de colores (red, white, black, etc).

El módulo de color de CSS3 incrementa las codificaciones de colores y permite utilizar **HSL** (Hue, Saturatio, Lightness), **CMYK** (Cyan, Magenta, Yellow, Black), **HSLA** (Hue, Saturation, Lightness, Apha) y **RGBA** (Red, Green, Blue, Apha). 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 offset 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

<https://imagecolorpicker.com/es>

<https://www.site24x7.com/es/tools/selector-de-codigo-color.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
0.7
0.5
0.3
0.2
Texto
Texto
Texto
Texto
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>

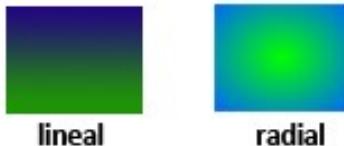
```

5 Gradienes

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



- **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> || <angulo>]? <stop>, <stop> [, <stop>]*)
- **-moz-radial-gradient(** [<posicion> || <angulo>]? [<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.
- **stop** 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.

Actualmente se necesitan cada vez menos utilizar los prefijos

Prefijo	Navegador
-moz-	Firefox
-webkit-	Safari y Chrome
-o-	Opera
-khtml-	Konqueror
-ms-	Internet Explorer y Microsoft Edge

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>.

6 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.

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, llenar 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 en WebKit (Chrome y Safari) han implementado una propiedad a partir de ésta.

Si se asigna un valor (en pixeles) 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:

6.1 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.

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

border-image: url(ficheroImagen) montajeImagen transformacionImagen

Donde

- **ficheroImagen**. Es la ruta de la imagen que se va a utilizar como borde, se pueden manejar hasta tres imágenes.
- **montajeImagen**. 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 pixeles. 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.

7 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
pulgadas (in). Una pulgada=2.54 cm. centímetros (cm). milímetros (mm). puntos (pt). Un punto=1/72 de pulgada. picas (pc). Una pica=12 puntos	em. Es relativo al tamaño de la M de la fuente instalada. ex: Es relativo a la altura de la x minúscula de la fuente utilizada. px. Es un píxel y es relativo a la resolución de la pantalla.

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>

7.1 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.

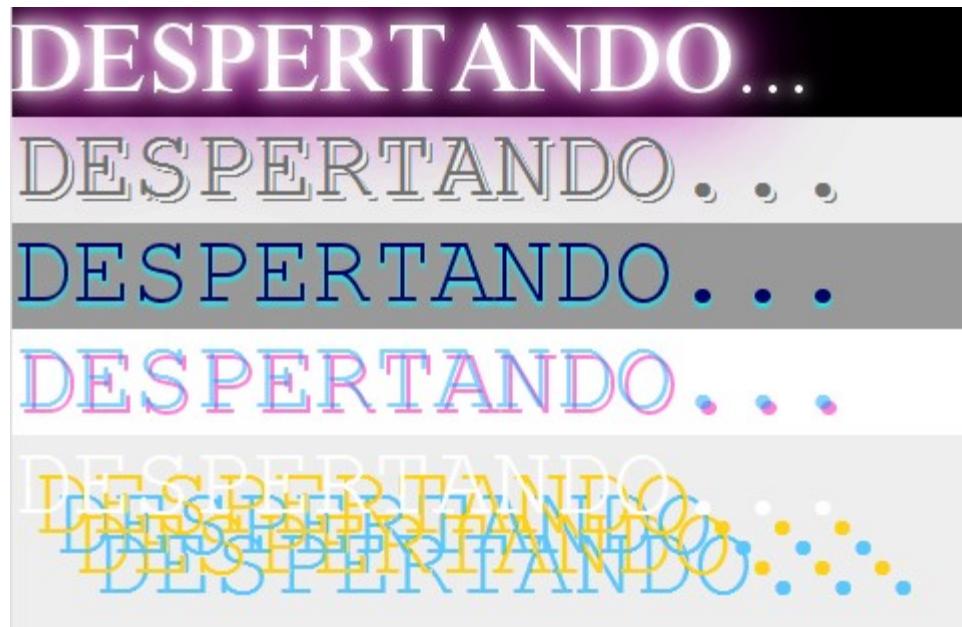
La siguiente imagen muestra distintos efectos logrados añadiendo sombras al texto.

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>
```