

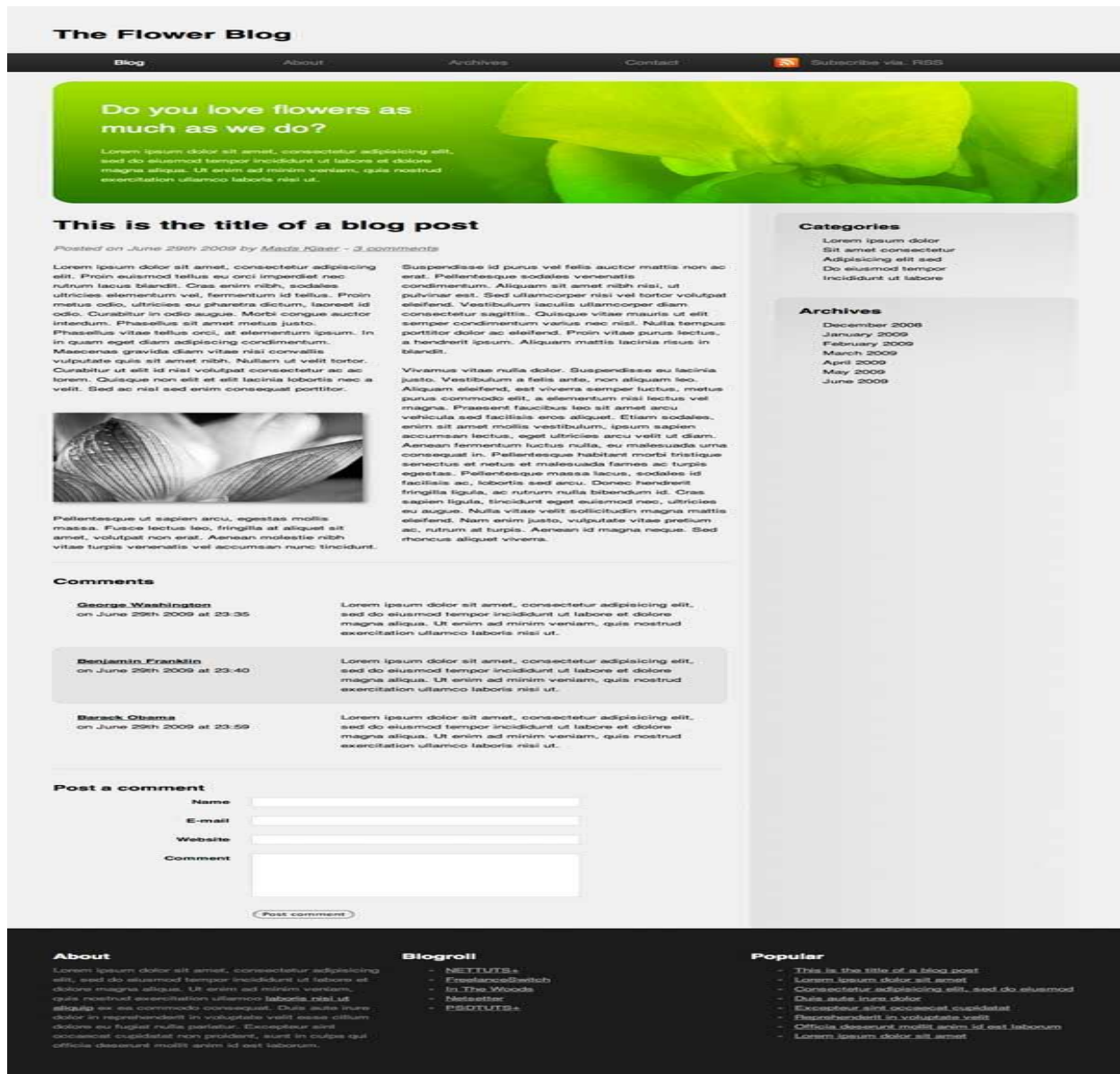
[HTML5 y CSS3: técnicas y tips para crear un blog de vanguardia](#)



Vamos a **construir una página o blog** utilizando las **técnicas** de la **próxima generación**, mediante **HTML 5** y **CSS3**, para estar siempre **actualizado** y listo para recibir a las **nuevas tendencias**.

Paso por paso explicaremos cómo construir cada uno de los **sitios** una vez que hayan sido **finalizadas las especificaciones** y **todos los navegadores** hayan **implementado** los cambios.

Lo que vamos a construir, Así lucirá nuestra página al terminarla:



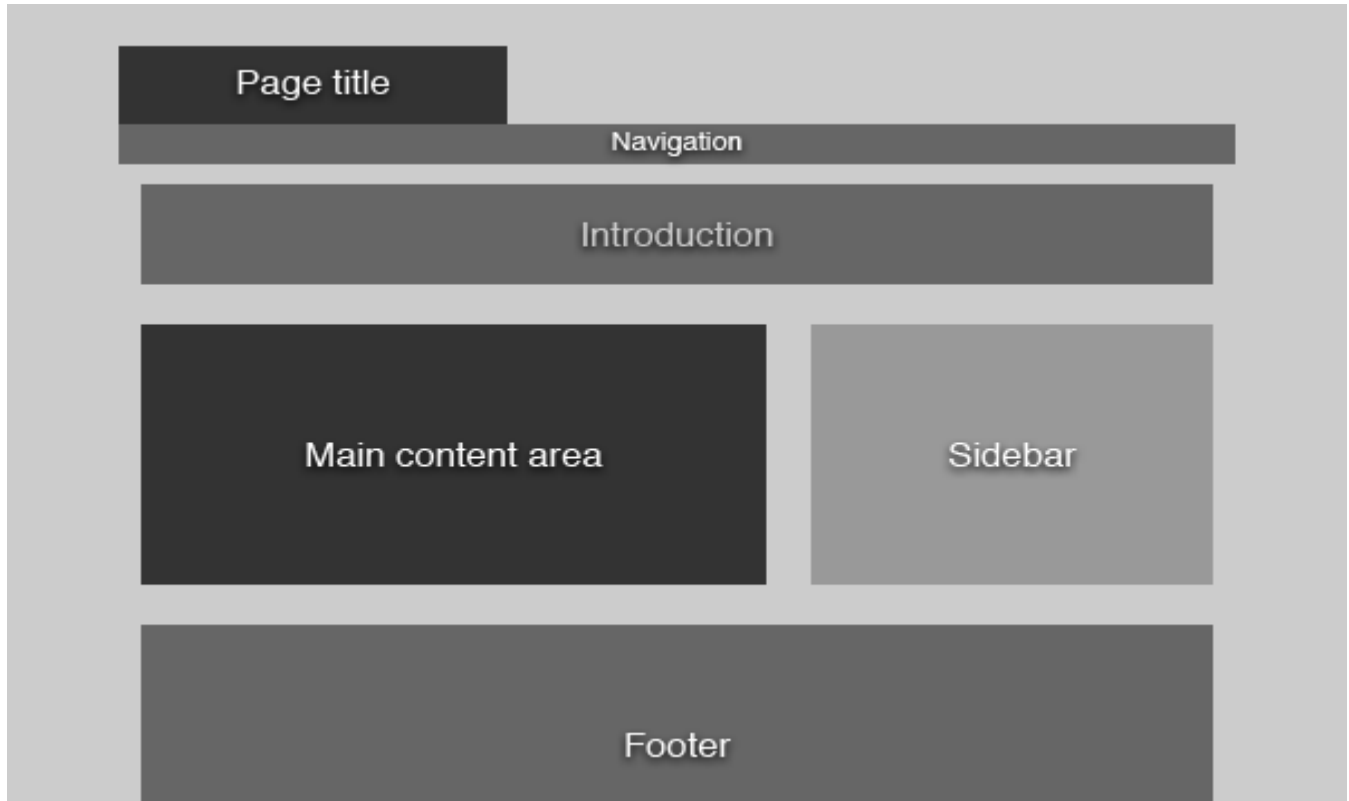
Algo bastante similar al diseño de tu página o blog de hoy en día. Cabecera con título, navegación horizontal, área de contenido con comentarios y formulario, sidebar y pie de página. Nada demasiado sofisticado. Comencemos a construirla.

HTML 5

HTML 5 es la siguiente versión mejorada de HTML. Posee una gran cantidad de elementos nuevos que hacen que nuestra página sea más semántica. Esto facilita el trabajo de los motores de búsqueda y los lectores de pantalla al navegar nuestras páginas, y mejoran la experiencia web de todos. Además, HTML 5 también incluye muy buenas APIs para dibujar gráficos en la pantalla, almacenar información offline, dragging and dropping, y mucho más.

Estructura básica

Antes de comenzar deberíamos tener la estructura básica:



En HTML 5 existen etiquetas específicas para delimitar la cabecera, el título, la navegación, la sidebar y el pie de página. Primero echémosle un vistazo al código y luego lo explicamos:

```
<!doctype html>
<html>
<head>
  <title>Pagina Final</title>
</head>
<body>
  <header>
    <h1>Pagina Final de Clases</h1>
  </header>
  <nav>
    <!-- Navigation -->
  </nav>
  <section id="intro">
    <!-- Introduction -->
  </section>
  <section>
    <!-- Main content area -->
  </section>
  <aside>
    <!-- Sidebar -->
  </aside>
  <footer>
    <!-- Footer -->
  </footer>
</body>
</html>
```

Todavía luce como código HTML, pero hay algunas cosas que debemos notar:

* En HTML 5, existe un sólo doctype. Está declarado al comienzo de la página mediante `<!doctype html>`. Y simplemente le informa al navegador que está lidiando con un documento HTML.

* La nueva etiqueta de título está envuelta en elementos introductorios, tales como el título de la página o un logo. Cada cabecera típicamente contiene una etiqueta heading de `<h1>` a `<h6>`. En este caso la cabecera se utiliza para dar introducción a la página completa, pero luego la utilizaremos para introducir una sección de la página.

* La etiqueta `nav` es utilizada para contener elementos navegacionales, tales como la navegación principal de un sitio o navegación más especializada como los links “anterior/siguiente”.

* La etiqueta `section` es usada para denotar una sección en el documento. Puede contener todo tipo de código y múltiples secciones pueden ser anidadas una dentro de la otra.

* La etiqueta `aside` es utilizada para envolver contenido relacionado con el contenido principal de la página, que podría ser separado del resto e igualmente tendría sentido. En este caso, lo estamos utilizando para la sidebar.

* La etiqueta `footer` debería contener información adicional sobre el contenido principal, como por ejemplo información del autor o del copyright, links a documentos relacionados, etc.

En lugar de utilizar `div`s para contener las diferentes secciones de la página, ahora estamos utilizando las etiquetas semánticas apropiadas.

Delimitando la navegación

La navegación es delimitada de la misma forma en que lo haríamos en HTML 4 o XHTML, utilizando una lista desordenada. La clave es que esta lista se ubica dentro de las etiquetas `nav`.

```
<nav>
  <ul>
    <li><a href="#">Inicio</a></li>
    <li><a href="#">Acerca de</a></li>
    <li><a href="#">Archivos</a></li>
    <li><a href="#">Contacto</a></li>
    <li class="subscribe"><a href="#">Suscríbese vía. RSS</a></li>
  </ul>
</nav>
```

Delimitando la introducción

Ya hemos definido una nueva sección en el documento mediante la etiqueta `section`. Ahora sólo necesitamos algo de contenido.

```
<section id="intro">
  <header>
    <h2>Te gustan las paginas asi como la hacemos?</h2>
  </header>
  <p> En HTML 5, existe un sólo doctype. Está declarado al comienzo de la página mediante
  <!doctype html>. Y simplemente le informa al navegador que está lidiando con un documento
  HTML.</p>
</section>
```

Añadimos un `id` a la etiqueta `section` para que la podamos identificar más tarde cuando llevemos a cabo el estilamiento. Utilizamos la etiqueta `header` para envolver todo lo que se encuentra alrededor del elemento introductorio `h2`. Además de describir un documento entero, la etiqueta `header` también debería usarse para describir las secciones individuales.

Delimitando el área de contenido principal

Nuestra área de contenido principal consiste de tres secciones: los posts del blog, los comentarios y el formulario de comentarios. Utilizando nuestros conocimientos sobre las nuevas etiquetas estructurales de HTML 5, debería ser fácil escribir el código.

Delimitando el post del Blog

Miremos el código y luego explicaremos los elementos:

```
<div id="content">
  <div id="mainContent">
    <section>
      <article class="blogPost">
        <header>
          <h2>Titulo Principal 2</h2>
          <p>Posteado el <time datetime="2009-06-29T23:31:45+01:00">June 29th 2009</time> por <a href="#">TecniredMX</a> – <a href="#comments">3 Comentarios</a></p>
        </header>
        <div>
          <p> En HTML 5, existe un s&ocute;lo doctype. Est&aacute; declarado al comienzo de la p&aacute;gina mediante <!doctype html>. Y simplemente le informa al navegador que est&aacute; lidiando con un documento HTML.&hellip;</p>
          
          <p> En HTML 5, existe un s&ocute;lo doctype. Est&aacute; declarado al comienzo de la p&aacute;gina mediante <!doctype html>. Y simplemente le informa al navegador que est&aacute; lidiando con un documento HTML.&hellip;</p>
          <p> En HTML 5, existe un s&ocute;lo doctype. Est&aacute; declarado al comienzo de la p&aacute;gina mediante <!doctype html>. Y simplemente le informa al navegador que est&aacute; lidiando con un documento HTML.&hellip;</p></div>
        </article>
      </section>
```

Comenzamos una nueva sección y envolvemos todo el post del blog en una etiqueta article. La etiqueta article se usa para denotar una entrada independiente en un blog, discusión, enciclopedia, etc. y es ideal para usarla aquí. Dado que estamos viendo los detalles de un sólo post, sólo tenemos un artículo, pero en la página principal del blog envolveremos cada post en una etiqueta article.

El elemento header es utilizado para presentar la cabecera y la información meta sobre el post. Le informamos a los usuarios cuándo fue escrito el post, quién lo escribió y cuantos comentarios tiene. La estampilla de tiempo está dentro de una etiqueta. Ésta también es nueva en HTML 5 y se usa para señalar un momento específico de tiempo. Los contenidos del atributo datetime deberían ser:



1. El año seguido por un guion del medio (-)
2. El mes seguido por un guion del medio (-)
3. La fecha
4. Una T mayúscula que denota que especificaremos el tiempo local
5. El tiempo local en el formato: hh:mm:ss
6. La zona horaria relativa al GMT.

Delimitando los comentarios

Delimitar los comentarios es bastante directo. No se necesita usar etiquetas ni atributos.

```

<section id="comments">
  <h3>Comentarios</h3>
  <article>
    <header>
      <a href="#">Felipe Calderon</a> el <time datetime="2009-06-29T23:35:20+01:00">June 29th
2009 at 23:35</time>
    </header>
    <p>Dos un conjunto m&acute;s amplio de tecnolog&iacute;as que permite a los sitios Web y a las
aplicaciones ser m&acute;s diversas y de gran alcance. Este conjunto se le llama HTML5 y amigos, a
menudo reducido a s&oacute;lo a HTML5.</p>
  </article>
  <article>
    <header>
      <a href="#">Vicente Fox</a> el <time datetime="2009-06-29T23:40:09+01:00">June 29th 2009 at
23:40</time>
    </header>
    <p>Dise&ntilde;ado para ser utilizable por todos los desarrolladores de Open Web, esta p&acute;gina
referencia a numerosos recursos sobre las tecnolog&iacute;as de HTML5, que se clasifican en varios grupos
seg&uacute;n su funci&oacute;n. </p>
  </article>
  <article>
    <header>
      <a href="#">Barack Obama</a> el <time datetime="2009-06-29T23:59:00+01:00">June 29th 2009
at 23:59</time>
    </header>
    <p>Uno se trata de una nueva versi&oacute;n del lenguaje HTML, con nuevos elementos, atributos y
comportamientos. </p>
  </article>

```

</section>

Delimitando el formulario de “ingresa tu comentario”

En HTML 5 se han introducido varias mejoras en lo que respecta a formularios. Ya no tenemos que hacer la validación del lado del cliente de los campos, mails requeridos, etc, sino que el navegador se encarga de eso por nosotros.

```
<form action="#" method="post">
  <h3>Haz un Comentario</h3>
  <p>
    <label for="nombre">Nombre</label>
    <input name="nombre" id=" nombre " required="" type="text">
  </p>
  <p>
    <label for="correo">Correo</label>
    <input name="correo" id="correo" required="" type="email">
  </p>
  <p>
    <label for="pagina">Página</label>
    <input name="pagina" id="pagina" type="url">
  </p>
  <p>
    <label for="comentario">Comentario</label>
    <textarea name="comentario" id="comentario" required=""></textarea>
  </p>
  <p><input value="Post comment" type="submit"></p>
</form>
</div>
```

Ahora hay dos tipos nuevos de entradas: e-mail y URL. E-mail especifica que el usuario debería ingresar una dirección de correo electrónica válida y, URL que el usuario debería ingresar una dirección de un sitio web válida. Si escribes required como atributo, el usuario no puede enviar un campo incompleto. “Required” es un atributo booleano, nuevo para HTML 5. Esto quiere decir que el atributo no se puede declarar sin ningún valor.

Delimitando la Sidebar y el pie de página

El código de la sidebar y el pie de página son extremadamente simple. Algunas secciones con algo de contenido dentro, el aside apropiado y las etiquetas footer.

```
<aside>
  <section>
    <header>
      <h3>Categorías</h3>
    </header>
    <ul>
      <li><a href="#">Sistemas</a></li>
      <li><a href="#">Informática</a></li>
      <li><a href="#">Descargas</a></li>
      <li><a href="#">Películas</a></li>
      <li><a href="#">Música</a></li>
    </ul>
  </section>
```

```

<section>
  <header>
    <h3>Archivos</h3>
  </header>
  <ul>
    <li><a href="#">Enero 2015</a></li>
    <li><a href="#">Febrero 2015</a></li>
    <li><a href="#">Marzo 2015</a></li>
    <li><a href="#">Abril 2015</a></li>
    <li><a href="#">Mayo 2015</a></li>
  </ul>
</section>
</aside>
</div>
<footer>
  <div>
    <section id="about">
      <header>
        <h3>Acerca de&hellip;</h3>
      </header>
      <p>Somos una empresa 100% mexicana que nos permitimos crear esta web para el apoyo de todos nuestros clientes, amistades y visitantes cibernautas que nos visitan dia a dia, toda la informaci&oacute;n que aqu&iacute; se encuentra es propia o una recopilaci&oacute;n de diferentes web que circulan por la internet, solo lo mejor esta aqu&iacute;.</p>
    </section>
    <section id="blogroll">
      <header>
        <h3>Enlaces</h3>
      </header>
      <ul>
        <li><a href="#">Google+</a></li>
        <li><a href="#">Facebook</a></li>
        <li><a href="#">Twitter</a></li>
        <li><a href="#">Youtube</a></li>
        <li><a href="#">Hotmail</a></li>
      </ul>
    </section>
    <section id="popular">
      <header>
        <h3>Populares</h3>
      </header>
      <ul>
        <li><a href="#">Pagina Oficial de Acu&ntilde;a Vive</a></li>
        <li><a href="#">Cecati 197 acu&ntilde;a</a></li>
        <li><a href="#">Tecnired mexico &ndash; Educacion de Calidad</a></li>
      </ul>
    </section>
  </div>
</footer>

```

Ahora pasemos al estilo.

Dando estilo con CSS3

CSS 3 está construido en base a los principios de estilos, selectores y cascadas que ya conocemos de la versión anterior de CSS. Pero añade muchas funcionalidades, incluyendo nuevos selectores, pseudo-clases y propiedades. Y gracias a estas nuevas funciones se vuelve mucho más fácil programar nuestro diseño.

Programación básica

Para comenzar vamos a definir algunas reglas básicas en lo que concierne a la tipografía, el color de fondo de la página, etc. Reconocerás todo esto de CSS 2.1

```
/* Reseteamos los estilos de margin y padding */
* {
    margin: 0;
    padding: 0;
}
/* Le dice al navegador que dibuje los elementos de HTML 5 como block */
header, footer, section, aside, nav, article {
    display: block;
}
body {
    margin: 0 auto;
    padding: 22px 0;
    width: 940px;
    font: 13px/22px Helvetica, Arial, sans-serif;
    background: #F0F0F0;
}
h1, h2 {
    font-size: 28px;
    line-height: 44px;
    padding: 22px 0;
}
h3 {
    font-size: 18px;
    line-height: 22px;
    padding: 11px 0;
}
p {
    padding-bottom: 22px;
}
```

Primero, reseteamos los estilos de margen y padding con una regla simple. Con esto bastará.

Luego le decimos al navegador que dibuje todos los nuevos elementos HTML 5 como un bloque. Los navegadores no tienen problemas con los elementos que no reconocen, pero no saben cómo deberían dibujar esos elementos por defecto. Debemos decirles esto hasta que el estándar sea implementado en todos lados.

Hasta que los navegadores implementen por completo HTML 5 y CSS 3 necesitaremos definir los valores en unidades relativas, por lo que es conveniente seleccionar el tamaño de la fuente en píxeles y no en ems o %.

Una vez hecho esto, continuemos.

Dándole estilo a la navegación

Es importante notar que el ancho del cuerpo ha sido definido como 940px y que fue centrado. Nuestra barra de navegación necesita hacer span en todo el ancho de la ventana, por lo que tendremos que aplicar algunos estilos adicionales:

```
nav {  
    position: absolute;  
    left: 0;  
    width: 100%;  
    background: url("images/nav_background.png");  
}
```

Posicionamos el elemento nav de forma absoluta, lo alineamos a la izquierda de la ventana y hacemos que haga span en todo el ancho. Centramos la lista anidada para mostrarla dentro de los límites del diseño:

```
nav ul {  
    margin: 0 auto;  
    width: 940px;  
    list-style: none;  
}
```

Ahora definiremos algunos estilos adicionales para hacer que los ítems de navegación luzcan más bonitos y los alinearemos dependiendo de nuestro diseño.

```
nav ul li {  
    float: left;  
}  
nav ul li a {  
    display: block;  
    margin-right: 20px;  
    width: 140px;  
    font-size: 14px;  
    line-height: 44px;  
    text-align: center;  
    text-decoration: none;  
    color: #777;  
}  
nav ul li a:hover {  
    color: #fff;  
}  
nav ul li.selected a {  
    color: #fff;  
}  
nav ul li.subscribe a {  
    margin-left: 22px;  
    padding-left: 33px;  
    text-align: left;  
    background: url("images/rss.png") left center no-repeat;  
}
```

Dándole estilo a la introducción

El código de la introducción es bastante simple: Una sección con un titular y un párrafo de texto. Sin embargo, usaremos algo de CSS 3 para hacerlo lucir más atractivo.

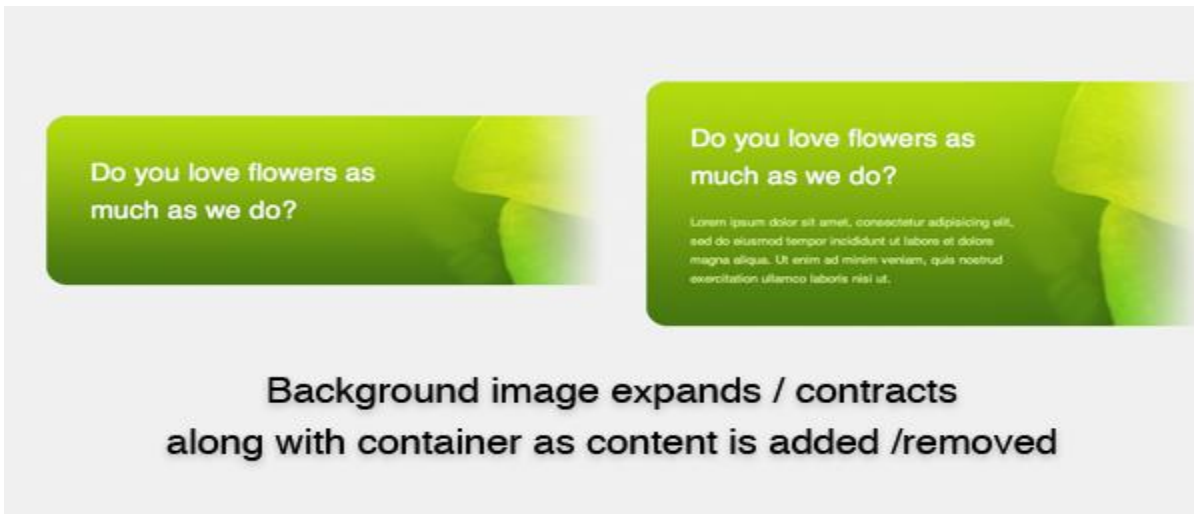
```
/*-----  
=INTRODUCTION
```

```

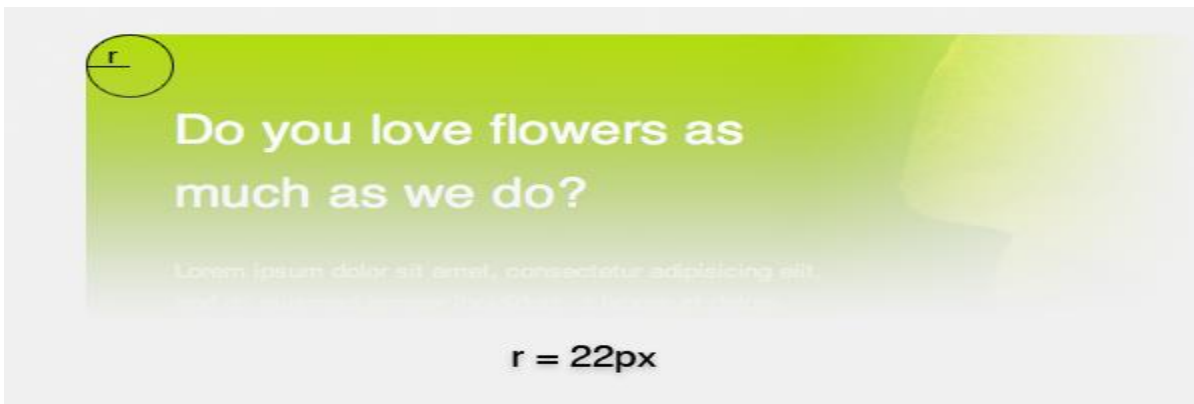
-----*/
#intro {
  margin-top: 66px;
  padding: 44px;
  background: #467612 url("intro_background.png") repeat-x;
  background-size: 100%;
  border-radius: 22px;
}

```

Estamos utilizando dos propiedades nuevas. La primera es background-size, que nos permite escalar la imagen de fondo. En nuestro caso, la escalamos al 100% en ambos ejes. Esto es algo que no se podía hacer en CSS 2.1 sin código no-semántico y sin tener varios problemas con el navegador.



La segunda propiedad nueva es border-radius, que aplica esquinas redondeadas al elemento. Puedes especificar valores diferentes para cada esquina o seleccionar sólo algunas esquinas redondeadas.



Desafortunadamente, ninguna de las propiedades son implementadas por completo en los navegadores. Pero a pesar de esto, podemos obtener algo de soporte utilizando atributos vendor-specific. Background-size es soportado por las nuevas versiones de Safari, Opera y Konqueror. Y border-radius es soportado por las nuevas versiones de Safari y Firefox. Realizando las correcciones debe quedar así.

```

#intro {
  position: relative;
  margin-top: 66px;
  padding: 44px;
  background: #467612 url("images/intro_background.png") repeat-x;

  /* Background-size not implemented yet */
}

```

```
-webkit-background-size: 100%;  
-o-background-size: 100%;  
-khtml-background-size: 100%;
```

```
/* Border-radius not implemented yet */  
-moz-border-radius: 22px;  
-webkit-border-radius: 22px;  
behavior:url(border.htc);  
border-radius: 22px;
```

```
}
```

Dado que tenemos un color de fondo definido, no habrá problemas mayores en los navegadores que no soporten la función background-size, como Firefox por ejemplo. Ahora sólo debemos darle estilo al titular y al texto.

```
#intro h2, #intro p {  
  position: relative;  
  z-index: 9999;  
  width: 336px;  
}
```

```
#intro h2 {  
  padding: 0 0 22px 0;  
  font-weight: normal;  
  color: #fff;  
}
```

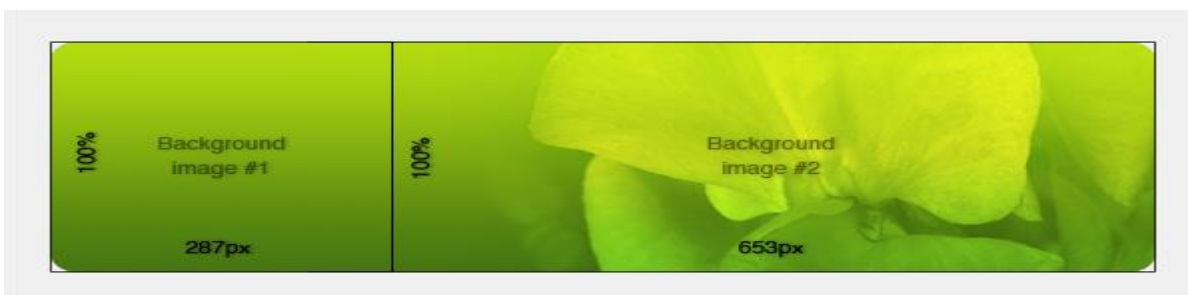
```
#intro p {  
  padding: 0;  
  color: #d9f499;  
}
```

```
#intro img {  
  position: absolute;  
  top: 0;  
  right: 0;  
  width: 653px;  
  height: 100%;
```

```
/* Border-radius not implemented yet */  
border-radius: 22px;  
-moz-border-radius: 22px;  
-webkit-border-radius: 22px;
```

```
}
```

Le damos a las dos imágenes de fondo dimensiones explícitas para asegurarnos que no se superpongan, y listo.



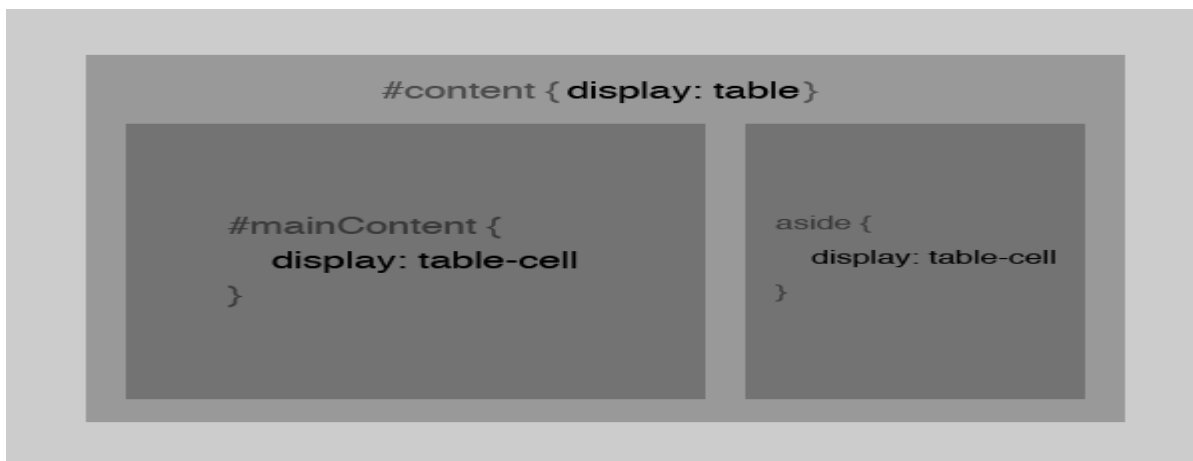
Dándole estilo al área de contenido y a la sidebar

El área de contenido y la sidebar serán alineadas una al lado de la otra. Tradicionalmente, haríamos esto mediante floats, pero en CSS lo haremos con tablas.

En CSS 3 podemos hacer que los elementos se comporten como tablas sin que esto se note en el código. Para comenzar, necesitaremos algunos divs para agrupar las secciones de una forma un poco más lógica.

```
<div id="content">
  <div id="mainContent">
    <section>
      <!-- Blog post -->
    </section>
    <section id="comments">
      <!-- Comments -->
    </section>
    <form>
      <!-- Comment form -->
    </form>
  </div>
  <aside>
    <!-- Sidebar -->
  </aside>
</div>
```

Todo sigue teniendo sentido semánticamente, pero ahora puedes darle estilo. Queremos que el div #content se comporte como una tabla, con #mainContent y aside como celdas de la tabla. Mediante CSS 3, esto es sencillo:



```
/*-----
=CONTENT AREA AND SIDEBAR LAYOUT
-----*/
#content {
  display: table;
}
#mainContent {
  display: table-cell;
  width: 620px;
  padding-right: 22px;
}
aside {
```

```

display: table-cell;
width: 300px;
background: url("images/sidebar_background.png") top left;
}

```

Dándole estilo al post del blog

Darle estilo al título del post es bastante trivial, así que pasaremos directamente a la parte divertida: el diseño de columnas múltiples.

Columnas múltiples

Las columnas múltiples de texto antes eran imposibles a menos que se separara el texto de forma manual, pero con CSS 3 es muy fácil. Aunque tendremos que añadir un div alrededor de los párrafos múltiples para que funcione en los navegadores actuales.

```

<div>
<p>Lorem ipsum dolor sit amet...</p>
<p>Pellentesque ut sapien arcu...</p>
<p>Vivamus vitae nulla dolor...</p>
...
</div>

```

Ahora podemos agregar dos propiedades simples:

```

.blogPost div {
column-count: 2;
column-gap: 22px;
}

```

Deseamos dos columnas y un espacio de 22px entre ellas. El div adicional se necesita porque actualmente no hay una forma soportada de hacer que un elemento tenga un span mayor que una columna. En el futuro, sin embargo, podremos especificar la propiedad span de la columna, y podremos escribir sólo:

```

/*-----
=BLOG POST
-----*/
.blogPost header p, .blogPost header p a {
font-size: 14px;
font-style: italic;
color: #777;
}

.blogPost header p a:hover {
text-decoration: none;
color: #000;
}

```

Por supuesto las propiedades column-count y column-gap son soportados sólo por algunos navegadores, Safari and Firefox. Así que por ahora debemos utilizar la propiedad vendor-specific.

```

.blogPost div {
/* Column-count not implemented yet */
-moz-column-count: 2;
-webkit-column-count: 2;

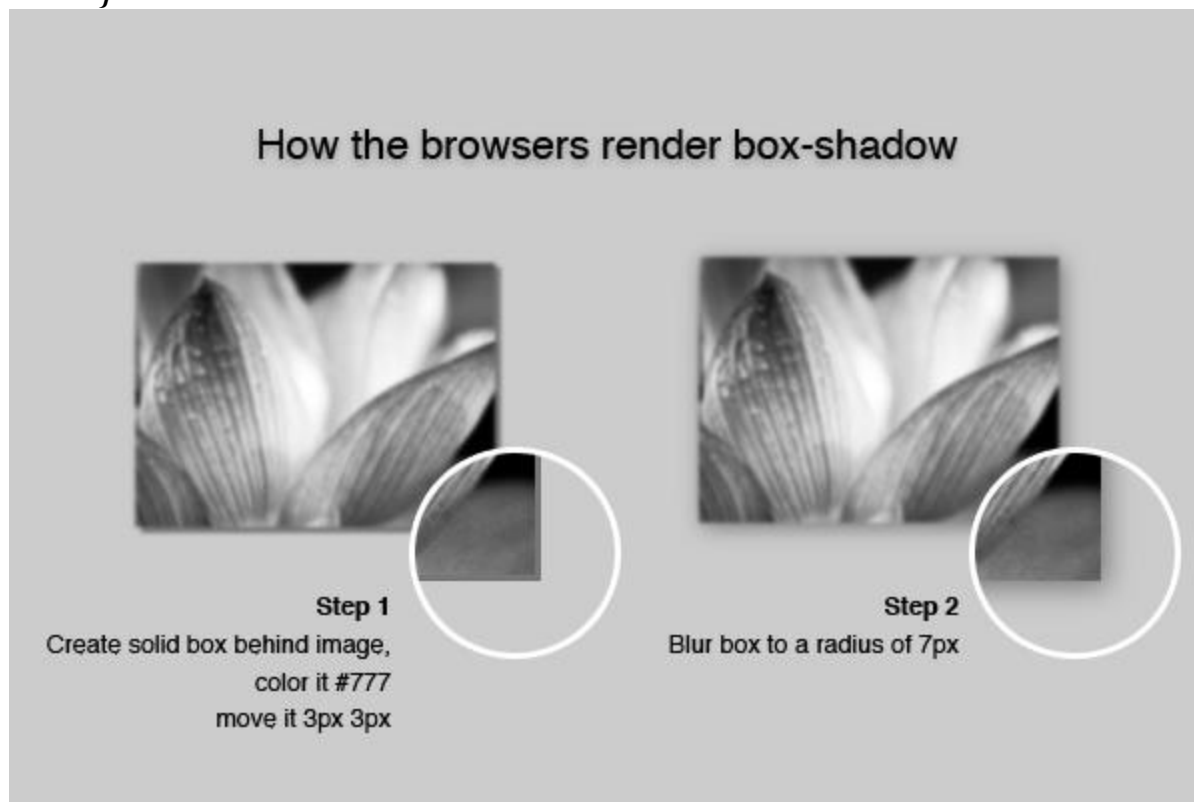
/* Column-gap not implemented yet */
-moz-column-gap: 22px;
-webkit-column-gap: 22px;
}

```

Sombra de caja (box-shadow)

Si miras con atención la imagen del post notarás una drop-shadow. Podemos generar esto mediante CSS 3 y la propiedad box-shadow.

```
.blogPost img {  
    margin: 22px 0;  
    -webkit-box-shadow: 3px 3px 7px #777;  
    border-radius: 22px;  
}
```



Los primeros “3px” le indican al navegador dónde queremos que la sombra pare horizontalmente. Los segundos “3px” le dicen dónde queremos que pare verticalmente. Los últimos “7px” indican cuán borroso debería ser el borde. Si lo programas en “0” será completamente sólido. Por último, definimos el color base de la sombra. El color se verá desvanecido por supuesto, dependiendo de cuanta borrosidad y sombra hayamos seleccionado.

No resulta sorprendente que esta propiedad todavía no haya sido implementada en todos los navegadores. De hecho, sólo funciona en Firefox 3 y Safari (aunque quizás también lo haga en Chrome, dado que ambos usan el motor Webkit). En ambos casos deberemos usar la propiedad vendor-specific.

Para Firefox 3, Safari y Chrome:

```
.blogPost img {  
    margin: 22px 0;  
    -webkit-box-shadow: 3px 3px 7px #777;  
    border-radius: 22px;  
}
```

Realizar Zebra-striping en los comentarios

Zebra-striping, o resaltar cada segundo elemento de una serie (como las rayas intercaladas de una cebra), ha involucrado tradicionalmente la selección de todos los elementos por medio de javascript, luego el loopeo a

través de ellos y finalmente el resaltado de todos los elementos impares. CSS 3 introduce la pseudo-clase “nth-child”, que nos permite llevar a cabo esta función de forma increíblemente fácil sin necesidad de javascript. Lo usaremos para hacer zebra-stripe sobre los comentarios.

```
/*-----  
=COMMENTS  
-----*/  
  
#comments {  
    margin-top: 21px;  
    padding-top: 22px;  
    border-top: 2px solid #d7d7d7;  
    border-radius: 10px;  
}  
  
#comments article {  
    display: table;  
    padding: 22px;  
}  
  
#comments article:nth-child(odd) {  
    padding: 21px;  
    background: #E3E3E3;  
    border: 1px solid #d7d7d7;  
    -moz-border-radius: 11px;  
    -webkit-border-radius: 11px;  
    border-radius: 11px;  
}  
  
#comments article header {  
    display: table-cell;  
    width: 220px;  
    padding-right: 22px;  
}  
  
#comments article header a {  
    display: block;  
    font-weight: bold;  
    color: #000;  
}  
  
#comments article header a:hover {  
    text-decoration: none;  
}  
  
#comments article p {  
    padding: 0;  
}
```

Dado que el estándar incluye los dos valores más usados como taquigrafía, par e impar. El resto del estilado de comentarios debería ser fácil de comprender con nuestros nuevos conocimientos.

Dándole estilo al formulario de comentario, el pie de página y la Sidebar

Deberemos reutilizar algunas de las técnicas de CSS3 aprendidas para dar estilo al formulario de “nuevo comentario”, al footer y a la sidebar. En el formulario de comentario y el footer, usaremos el mismo tipo de estilo de tabla que en el diseño principal. En la sidebar, en cambio, utilizaremos border-radius para agregar esquinas redondeadas a las diferentes secciones.

```
/*-----  
=COMMENT FORM  
-----*/  
form {  
    margin-top: 21px;  
    padding-top: 22px;  
    border-top: 1px solid #d7d7d7;  
}  
  
form p {  
    display: table;  
    margin-bottom: 22px;  
    padding: 0 22px;  
}  
  
form label {  
    display: table-cell;  
    width: 140px;  
    padding-right: 20px;  
    text-align: right;  
    font-weight: bold;  
    vertical-align: top;  
}  
  
form input[type="text"], form input[type="email"], form input[type="url"] {  
    display: table-cell;  
    width: 300px;  
    height: 20px;  
    border: 1px solid #d7d7d7;  
}  
  
form textarea {  
    width: 300px;  
    height: 100px;  
    border: 1px solid #d7d7d7;  
}  
  
form input[type="submit"] {  
    margin-left: 162px;  
}
```



```

/*-----
=SIDEBAR
-----*/
aside section {
    margin: 22px 0 0 22px;
    padding: 11px 22px;
    background: url("images/sidebar_section_background.png") repeat-x;

    /* Border-radius not implemented yet */
    -moz-border-radius: 11px;
    -webkit-border-radius: 11px;
    border-radius: 11px;
}

aside section ul {
    margin: 0 0 0 22px;
    list-style: none;
}

    aside section ul li a {
        display: block;
        text-decoration: none;
        color: #000;
    }

        aside section ul li a: hover {
            text-decoration: underline;
        }

```

```

/*-----
=FOOTER
-----*/
footer {
    position: absolute;
    left: 0;
    width: 100%;
    background: #222;
}

footer div {
    display: table;
    margin: 0 auto;
    padding: 44px 0;
    width: 940px;
    color: #777;
}

```

```

    footer div section {
        display: table-cell;
    }

```

```

        width: 300px;
    }

    footer div #about, footer div #blogroll {
        padding-right: 20px;
    }

    footer h3 {
        color: #FFF;
    }

    footer a {
        color: #999;
    }

    footer a:hover {
        color: #FFF;
        text-decoration: none;
    }

    footer ul {
        margin: 0 0 0 40px;
        list-style: square;
        color: #565656;
    }

    footer ul li a {
        display: block;
    }

```

El diseño final

Lo podrás ver con todo el estilo ya aplicado.

Compatibilidad

La página se verá correctamente en Safari 4 y en los nuevos navegadores basados en webkit, porque son los únicos motores que soportan todas las técnicas CSS3 que hemos utilizado. Firefox 3 tiene algunos problemas al aplicar las esquinas redondeadas y no soporta background-size, pero más allá de eso el diseño funciona. También podrías definir algunas reglas más y hacer que todo funcione en los mayores navegadores, pero eso excede nuestro tutorial.

Conclusión

Una vez que HTML 5 y CSS 3 sean implementados en todos los navegadores será mil veces más sencillo construir un sitio web de lo que ya es. Finalmente podremos dejar de utilizar floats para los diseños y pasaremos un tiempo considerablemente menor escribiendo javascript para poder escalar nuestras imágenes de fondo o hacer zebra-stripe en nuestras tablas. ¡Esperemos que esto suceda rápido!

Este manual fue adaptado y se han realizado algunas modificaciones por www.tecniredmx.com

Fuente: Elwebmaster.com