

GUIA de HTML5

¿Qué es HTML?

HTML (*HyperText Markup Language*) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces (hyperlinks)* que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones multimedia* (gráficos, sonido...) La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por programas especializados conocidos como Browsers o navegadores web (Firefox,Opera,Chrome,etc.)

Estructura Básica de un documento HTML

Un documento escrito en HTML contendría básicamente las siguientes etiquetas:

<HTML> Indica el inicio del documento.
<HEAD> Inicio de la cabecera.
<TITLE> Inicio del título del documento.
</TITLE> Final del título del documento.
</HEAD> Final de la cabecera del documento.
<BODY> Inicio del cuerpo del documento.
</BODY> Final del cuerpo del documento.
</HTML> Final del documento.

¿Qué es HTML5?

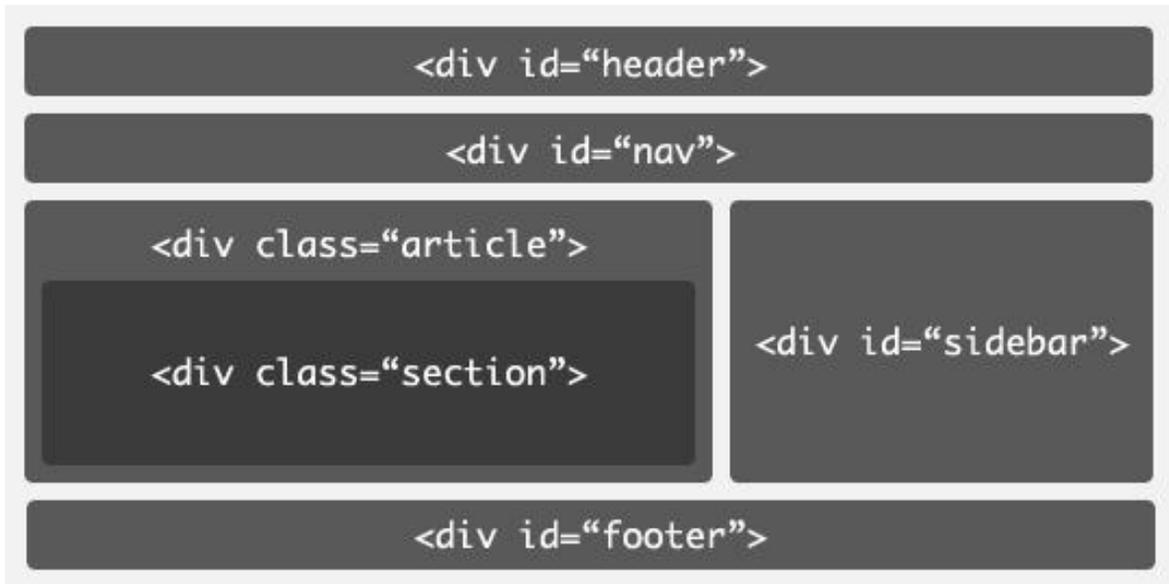
Es la quinta revisión importante del estándar que mueve internet, HTML y que ha sido impulsado por el aumento de las necesidades de mejorar la funcionalidad y la interactividad de usuarios y aplicaciones. **HTML 5** presenta una amplia gama de mejoras en controles para formularios, **APIs**, multimedia y también en la **estructura** y la **semántica**. El trabajo en **HTML 5** comenzó en el 2004 y es un esfuerzo conjunto entre el **W3C HTML WC** y el **WHATWG**. Muchos aportan su participación, como Apple, Mozilla, Opera, Microsoft y un sinnúmero de personas y empresas.

La aparición de estas revisiones implicará cambios importantes en el desarrollo de páginas web y algunos se aventuran incluso a prever la desaparición de algunas tecnologías de gran calado en la red, como puede ser Adobe Flash. Cierto o no, la realidad es que estamos presenciando una evolución radical de los lenguajes principales de la web que aportará grandes mejoras, pero que al mismo tiempo implicará un periodo de adaptación y aprendizaje por parte de programadores y diseñadores

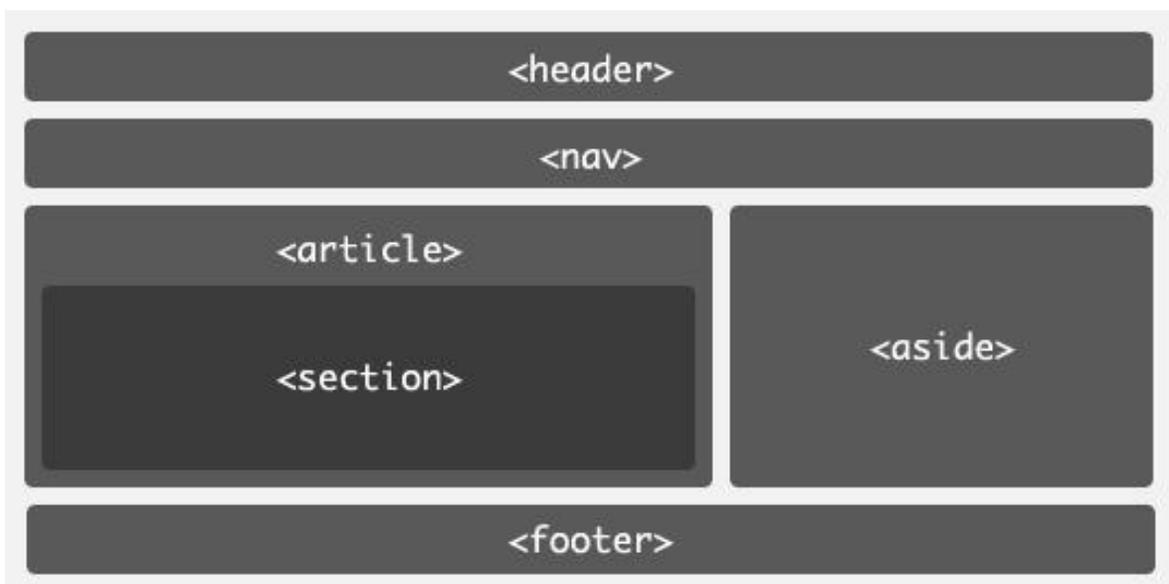
Novedades de la estructura:

Para crear una página web, normalmente se incluyen estructuras comunes como headers, footers y columnas y es común usar divs para darles un id descriptiva o clase ya que las actuales versiones de HTML4 carecen de la semántica necesaria para describir estas partes de manera específica.

Por ejemplo:



El HTML5 soluciona esto incluyendo nuevos elementos que representan cada una de las diferentes secciones de una página web.



Nuevos elementos:

Los tiempos modernos requieren nuevos elementos para proporcionar una web más semántica, completa y homogénea. Para ello se han añadido una buena serie de elementos que nos permitirán encapsular más nuestro contenido.

- `<article />` elemento que nos permite declarar un trozo del contenido como artículo. Ideal para blogs o periódicos.
- `<aside />` representa un trozo de contenido que se relaciona muy levemente con el resto del contenido.
- `<dialog />` elemento que permite representar conversaciones:

```
<dialog>
<dt> Costello
<dd> Look, you gotta first baseman?
<dt> Abbott
<dd> Certainly.
<dt> Costello
<dd> Who's playing first?
<dt> Abbott
<dd> That's right.
<dt> Costello
<dd> When you pay off the first baseman every month, who gets the money?
<dt> Abbott
<dd> Every dollar of it.
</dialog>
```

- `<figure />` podrás usarlo para asociar con un caption un contenido incrustado, como por ejemplo gráficos o vídeo:

```
<figure>
  <video src=ogg>...</video>
  <legend>Example</legend>
</figure>
```

- `<footer />` Sección de la página para contener información sobre el autor, copyright, etc,...
- `<header />` representa a la sección de cabecera.
- `<nav />` representa la sección de la página orientada a la navegación.
- `<section />` elemento que indica que se trata de una sección genérica.
- `<audio />` y `<video />` para el contenido multimedia.
- `<embed />` es un elemento dedicado para contenido de plugins.
- `<m />` representa el texto marcado.
- `<meter />` usado para representar medidas, por ejemplo el tamaño del disco usado...
- `<time />` usado para mostrar fechas y/o tiempo.
- `<canvas />` usado para mostrar gráficos renderizados en tiempo real, por ejemplo gráficos, juegos, etc,...

- <command /> relacionado con los comandos que el usuario puede invocar.
- <datagrid /> ideal para mostrar un arbol de datos o una tabla tabulada.
- <details /> muestra información adicional si el usuario lo demanda.
- <datalist /> junto con el nuevo atributo list para los <input /> puede ser usado para crear

comboboxes:

```
input list=browsers>
<datalist id=browsers>
  <option value="Safari">
  <option value="Internet Explorer">
  <option value="Opera">
  <option value="Firefox">
</datalist>
```

- <event-sources /> puede ser usado para capturar eventos enviados desde servidor.
- <output /> nos indica que tipo de salida vamos producir con nuestra página.
- <progress /> representa una barra de proceso de una tarea, por ejemplo descargar,...

Los elementos de entrada <input /> dispondrán de una serie de tipos (type) nuevos para indicar los diferentes tipos de elementos de entrada posibles.

- datetime
- datetime-local
- date
- month
- week
- time
- number
- range
- email
- url

La idea es que estos tipos sean proporcionados por el agente de usuario (navegador) en su interface que submittarán el formato definido al servidor.

Nuevos atributos:

- media: Para conseguir una mayor consistencia con el link en los elementos <a />
- ping: Especificaremos una lista separada por espacios donde produciremos un ping cuando se siga el enlace, para los elementos <area /> y <a />
- target: Disponible para mejorar la consistencia con el elemento <a />.
- autofocus: Destinado para indicar el elemento <input /> (no hidden), <select />, <textarea /> o <button /> que ha de coger el foco al cargar la página.
- form: Atributo para <input />, <output />, <select /> <textarea />, <button /> y <fieldset /> que permite que se asocien con un simple formulario.
- replace: atributo para <input />, <button /> y <form /> que le afectará cuando el contenido del elemento sufra algún cambio.
- data: Para <form />, <select /> y <datalist />.

- required: Para elementos <input /> (Excepto hidden e image) y <textarea />, indica que el campo es obligatorio.
- inputmode: Atributo para <input /> y <textarea />.
- disabled: Para <fieldset />, permite desactivar el fieldset completo.
- autocomplete, min, max, pattern, step: Para elementos <input /> permite delimitar las posibilidades de nuestros elementos de entrada.
- list: Para elementos <datalist /> y <select />.
- template: Para <input /> y <button /> podrá usarse para repetir templates.
- scoped: Para elemento <style />, permitirá usar hojas de estilo "scoped" ??
- async: Para el elemento <script /> el ajax hecho atributo.

Elementos Cambiados

Estos elementos de HTML5 son incompatibles con HTML4.

- El elemento <a /> sin href ahora creará un enlace al sitio.
- El elemento <address /> es ahora un nuevo concepto de sección.
- El elemento ahora representa un trozo de texto a ser estilizado sin ninguna importancia.
- Para elementos <label /> el navegador no debe mover el foco desde la etiqueta al control a menos que el comportamiento sea estandar para el interfaz utilizado en la plataforma.
- <menu /> ha sido redefinido para ser usado con los actuales menús.
- El elemento <small /> ahora representa una impresión pequeña.
- El elemento definitivamente representa el énfasis puesto en trozo de nuestro texto.

Elementos eliminados

En la nueva versión, algunos de los elementos anteriormente desaprobados pasan a ser eliminados definitivamente.

- acronym
- applet
- basefont
- big
- center
- dir
- font
- frame
- frameset
- isindex
- noframes
- noscript (solo en XHTML5)
- s
- strike
- tt
- u

Atributos eliminados

Al igual que los elementos los atributos también pasarán a mejor vida.

- rev y charset en <link /> y <a />
- target en <link />
- nohref en <area />
- profile en <head />
- version en <html />
- name en <map />
- scheme en <meta />
- archive, classid, codetype, declare y standby en <object />
- valuetype en <param />
- charset en <script />
- summary en <table />
- header, axis y abbr en <td /> y <th />

La nuevas propiedades de CSS3

El W3C permite efectuar nuevas acciones sobre los elementos de HTML, mediante nuevas propiedades, veamos qué cosas nuevas se pueden hacer con estas propiedades:

Nota: Aquí utilizo el prefijo -moz- para Mozilla Firefox. Existen otros prefijos para Chrome (-webkit), Safari (-webkit) y Opera (-o), basta con sustituir el prefijo para que funcione con el navegador respectivo

Borde con colores diferentes

La propiedad -moz-border-colors: permite crear varios bordes de colores diferentes. Esta propiedad puede ser utilizada también como:

-moz-borders-top-color: (adición de bordes, top, bottom, left, right)

Ej:

```
#midiv {
  border: 8px solid #000000;
  -moz-border-colors : #CC0000 #BB0000 #AA0000 #990000 #880000 #770000 #660000
  #550000;
  padding: 5px ;
}
```



Mozilla/Firefox users should see a nicely fading red border on this box...

□ Imágenes como bordes

CSS3 permite el uso de imágenes como bordes de los elementos de la página. Las dos propiedades (y sus derivadas) son:

border-image:

border-top-image
border-right-image
border-bottom-image
border-left-image

border-corner-image:

border-top-left-image border-top-right-image border-bottom-left-image border-bottom-right-image

Ej:

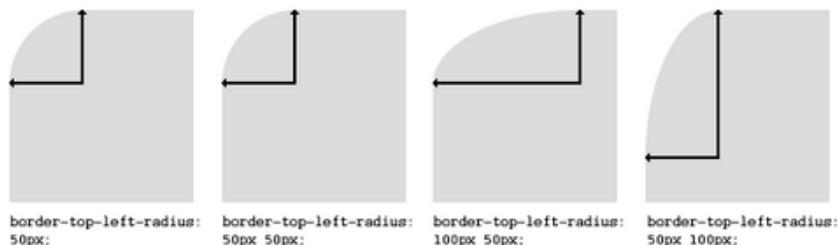
```
#mi-div {  
  border-image: url(border.png) 27 27 27 27 round round;  
}
```



□ Bordes redondeados en las esquinas

La propiedad **border-radius** de CSS3 permite a los desarrolladores web definir bordes redondeados en las esquinas, sin necesidad de imágenes de esquinas ni recurrir al uso de etiquetas div multiples.

```
#contenedor {  
  -moz-border-radius: 15px; /* Prefijo Moz para Mozilla (no valido para la W3C)*/  
  border-radius: 15px; /* Ningún prefijo para los navegadores que incorporan la propiedad sin prefijo (valido para la W3C) */  
}
```



□ Crear sombras en CSS3

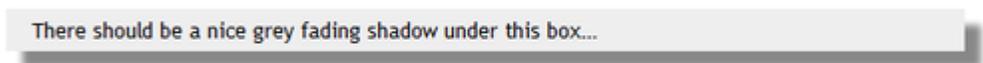
Una nueva funcionalidad de CSS3 implementada a partir de la versión 3.1 de Firefox, es la posibilidad de crear sombras de colores: esta es la propiedad **box-shadow**

Esta propiedad requiere de algunos parámetros para definir las características de la sombra:

1. Desplazamiento horizontal de la sombra: un valor positivo significa que la sombra aparece desde la derecha, un desplazamiento negativo hará que la sombra aparezca desde la izquierda.
2. Desplazamiento vertical de la sombra: un valor negativo significa que el box-shadow aparecerá desde arriba, un valor negativo hará aparecer la sombra desde abajo.
3. En cuanto al difuminado, cuanto más cerca de cero esté este valor, la sombra será más definida. En cambio, cuanto más se acerque de 1, la sombra estará más difuminada.

Ejemplos:

```
.sombra {  
  box-shadow: 10px 10px 5px #888;  
  padding: 5px 5px 15px;  
}
```



There should be a nice grey fading shadow under this box...

□ Imágenes de fondo de elementos en CSS3

1. Propiedades background-clip y background-origin

La propiedad **background-origin** de CSS3 permite determinar la manera en que la imagen de fondo se posicionará en un elemento, y la propiedad **background-clip** permite seleccionar una porción de la imagen

Ambas propiedades pueden tomar alguno de los 3 valores: **border-box**, **padding-box** y **content-box**.

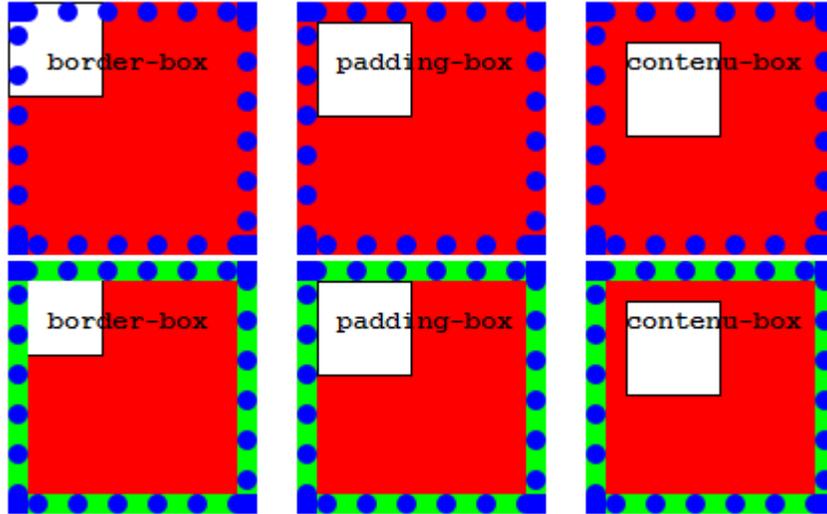
Las implementaciones experimentales tienen como propiedades:

```
-webkit-background-origin: padding-box;  
-moz-background-origin: padding-box
```

```
-webkit-background-clip: border-box;  
-moz-background-clip: border-box
```

Las implementaciones estables

(oficiales): `background-origin: content-box;`
`background-clip: content-box;`



2. Propiedad background-size

CSS3 permite especificar un tamaño a las imágenes de fondo. Este tamaño puede ser especificado en píxeles, (height y width), o en porcentaje. Si se especifica una tamaño en porcentaje, el tamaño es relativo al ancho o altura de la zona a la que se ha atribuido la propiedad **background-origin**.



La primera imagen contiene un fondo que abarca todo el contenedor, mientras q la segunda especifica el tamaño del fondo sobre el contenedor

3. Imágenes de fondo múltiples en CSS3

Es simple, coloca los “backgrounds” separados por coma y asígnale las propiedades respectivas a cada uno

```
body {  
    background: url(paisaje.png) top left no-repeat, url(oveja.jpg) bottom center  
    100px no-repeat;  
}
```

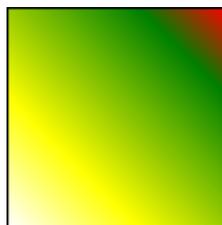


Degradados

Con la nueva propiedad **gradient** podemos crear degradados de varios colores sin necesidad de utilizar imágenes!

La sintaxis de la propiedad depende de los navegadores, aquí algunos ejemplos:

```
.gradient {  
    background: linear-gradient(1 225deg, 2 red, 3 green 20%, 4 yellow  
70%, 5 white);  
}
```



Animaciones y Transformaciones con CSS3

1. Transiciones

Su característica es cambiar una propiedad (o un grupo de ellas) en un período de tiempo determinado. Una ventaja por sobre **Javascript** es su degradación, ya que si esta propiedad no es soportada por el browser, la animación simplemente no es mostrada.

El siguiente código muestra el aumento del ancho de un **<div>** en un segundo cuando el *mouse* se posiciona sobre él. Luego, cuando el mouse sale del **<div>** su ancho vuelve a la posición inicial en 2 segundos:

```
.prueba1 { margin:10px;
width:200px; height:200px;
background:#f00;
border:1px solid #666;
-webkit-transition: all 1s ease; /* Safari-Chrome */
}

.prueba1:hover{
width:400px;
-webkit-transition: all 2s ease; /* Safari-Chrome */
}
```

Propiedades de transition:

transition-property: Determina la(s) propiedad(es) a ser animadas; puede ser cualquier propiedad CSS.

transition-duration: Indica la duración de la animación del inicio al fin en segundos. Por defecto el valor es 0.

transition-timing-function: Definen física de animación: ease (por defecto), linear, ease-in, ease-out y ease-in-out.

transition-delay: Tiempo en que la animación debe pausarse antes de comenzar.

2.

Transformaciones:

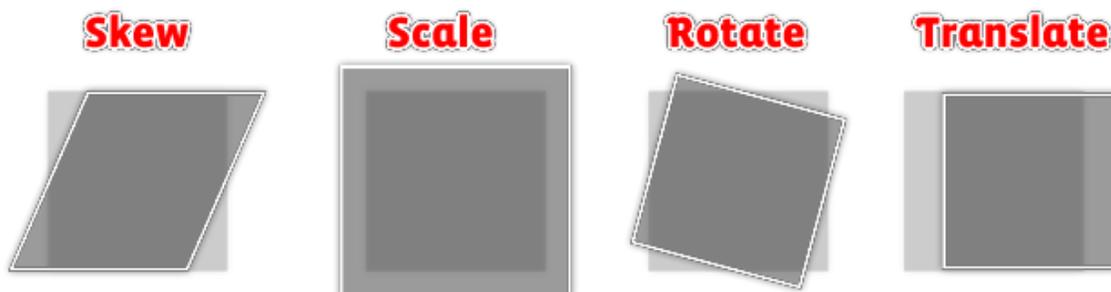
Existen 4 tipos de valores para transformar elementos HTML mediante CSS3:

Skew: desplazamiento de los ejes horizontales.

Scale: modificación de la escala del elemento.

Rotate: cambio de rotación del mismo definido en grados.

Translate: desplazamiento del elemento desde su posición original.



Practica 1: elaboración de “sticky notes” de colores con CSS3

En esta práctica construiremos una lista de HTML en un grupo de “sticky notes” o papeles adhesivos, de esos que se utilizan en oficinas.

El efecto es soportado en los navegadores modernos actualizados. Para los navegadores antiguos se pueden visualizar algunas de las propiedades.

RESULTADO FINAL



Paso 1: Construyendo el esqueleto y los cuadrados de los “stickys”

Arrancamos diseñando lo que sería la estructura básica de HTML que puede ser vista en todos los navegadores. Los sticky’s estarán representados en una lista no ordenada (o ‘ul’) que será modificada con estilos de CSS3.

En este paso se deben realizar 2 archivos: **sticky.html**, que contiene la estructura en HTML y **estilo.css**, para guardar los estilos CSS3. Los dos archivos deben ser guardados en la misma carpeta

A continuación se presenta el código de cada uno.

sticky.html

```
<!DOCTYPE html>
<html lang='es'>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Paso 1 – Sticky Notes</title>
  <link rel="stylesheet" href="estilo.css"/>
</head>
<body>
  <ul>
    <li>
      <a href="#">
        <h2>Titulo #1</h2>
        <p>Texto de contenido #1</p>
      </a>
    </li>
    <li>
      <a href="#">
        <h2>Titulo #2</h2>
        <p>Texto de contenido #2</p>
      </a>
    </li>
    <li>
      <a href="#">
        <h2>Titulo #3</h2>
        <p>Texto de contenido #3</p>
      </a>
    </li>
    <li>
      <a href="#">
        <h2>Titulo #4</h2>
        <p>Texto de contenido #4</p>
      </a>
    </li>
    <li>
      <a href="#">
        <h2>Titulo #5</h2>
        <p>Texto de contenido #5</p>
      </a>
    </li>
    <li>
      <a href="#">
        <h2>Titulo #6</h2>
        <p>Texto de contenido #6</p>
      </a>
    </li>
  </ul>
</body>
</html>
```

estilo.css

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

body{
  font-family:arial,sans-serif;
  font-size:100%;
  margin:3em;
  background:#666;
  color:#fff;
}

h2,p{
  font-size:100%;
  font-weight:normal;
}

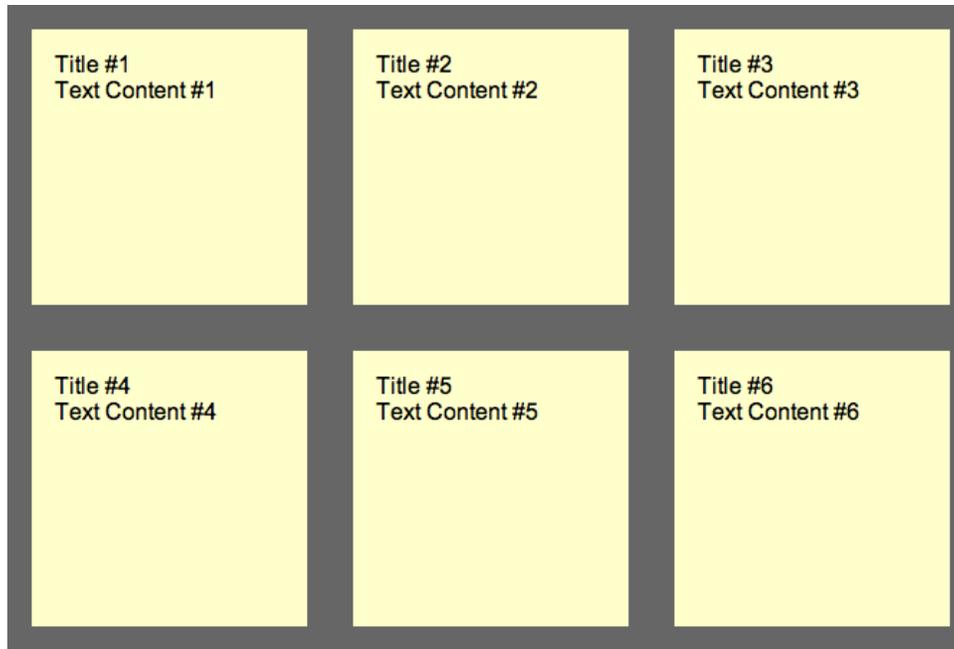
ul,li{
  list-style:none;
}

ul{ overflow:hidden;
  padding:3em;
}

ul li a{
  text-decoration:none;
  color:#000;
  background:#ffc;
  display:block;
  height:10em; width:10em;
  padding:1em;
}

ul li{
  margin:1em;
  float:left;
}
```

El resultado obtenido debe ser similar a la siguiente imagen:



Paso 2: Agregando sombras y fuentes personalizadas

Para darle más vida a los cuadrados que acabamos de relizar le agregaremos un par de sombras y haremos un cambio en la fuente utilizando el servicio gratuito Google Web Fonts.

Debajo del código donde agregamos el tag 'link' a nuestro estilo (**estilo.css**), introducimos el siguiente código HTML:

```
<link href="http://fonts.googleapis.com/css? family=Reenie+Beanie:regular" rel="stylesheet" type="text/css">
```

Estas líneas permiten comunicarnos de forma gratuita al servicio de fuentes de tipografía para la web de Google y utilizar la fuente que escojamos. En este caso, escogemos la fuente "Reenie Beanie".

El resultado de dicho cambio produce el siguiente archivo:

Nota: las líneas subrayadas en verde indican los cambios que fueron realizados en este paso.

sticky.html

```
<!DOCTYPE html>
<html lang='es'>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Paso 1 – Sticky Notes</title>
  <link rel="stylesheet" href="estilo.css"/>
  <link href="http://fonts.googleapis.com/css? family=Reenie+Beanie:regular" rel="stylesheet" type="text/css">
</head>
```

```

<body>
  <ul>
    <li>
      <a href="#">
        <h2>Titulo #1</h2>
        <p>Texto de contenido #1</p>
      </a>
    </li>
    <li>
      <a href="#">
        <h2>Titulo #2</h2>
        <p>Texto de contenido #2</p>
      </a>
    </li>
    <li>
      <a href="#">
        <h2>Titulo #3</h2>
        <p>Texto de contenido #3</p>
      </a>
    </li>
    <li>
      <a href="#">
        <h2>Titulo #4</h2>
        <p>Texto de contenido #4</p>
      </a>
    </li>
    <li>
      <a href="#">
        <h2>Titulo #5</h2>
        <p>Texto de contenido #5</p>
      </a>
    </li>
    <li>
      <a href="#">
        <h2>Titulo #6</h2>
        <p>Texto de contenido #6</p>
      </a>
    </li>
  </ul>
</body>
</html>

```

Ahora modificamos el archivo **estilo.css** para agregar el soporte a la fuente de Google y agregarle la sombra a los sticky notes. El archivo **estilo.css** quedará de la siguiente forma:

estilo.css

```

*{
  margin:0;
  padding:0;
}

```

```

body{

```

```
font-family:arial,sans-serif;
font-size:100%;
margin:3em;
background:#666;
color:#fff;
}

h2,p{
font-size:100%;
font-weight:normal;
}

ul,li{
list-style:none;
}

ul{ overflow:hidden;
padding:3em;
}

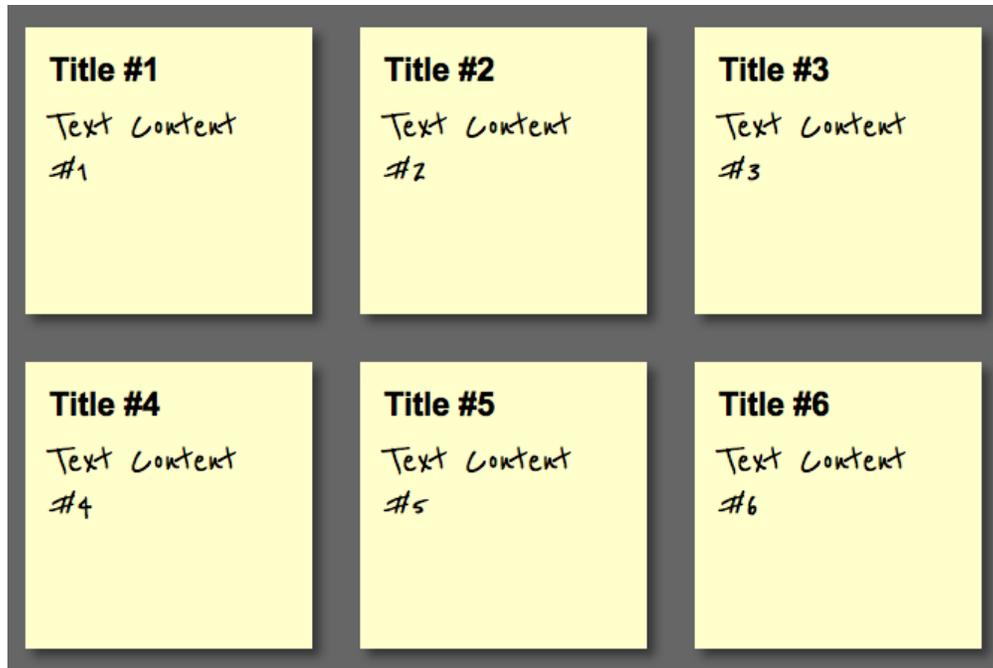
ul li a{
text-decoration:none;
color:#000;
background:#ffc;
display:block;
height:10em; width:10em;
padding:1em;
-moz-box-shadow:5px5px7pxrgba(33,33,33,1); /*Firefox*/
-webkit-box-shadow:5px5px7pxrgba(33,33,33,.7); /*Safari+Chrome*/
box-shadow:5px5px7pxrgba(33,33,33,.7); /*Opera*/
}

ul li{
margin:1em;
float:left;
}

ul li h2{
font-size:140%;
font-weight:bold;
padding-bottom:10px;
}

ul li p{
font-family:"Reenie Beanie",arial,sans-serif;
font-size:180%;
}
```

Para este paso, el resultado final sería el siguiente:



Paso 3: Rotando los Sticky notes

En este paso agregamos uno de los nuevos efectos que nos ofrece la tecnología CSS3: rotaciones.

Las rotaciones en CSS3 son consideradas como "Transiciones", y poseen una sintaxis en la que podemos especificar el número de grados a los cuales queremos rotar un elemento.

Únicamente cambiaremos el archivo **estilo.css**, obteniendo el siguiente contenido:

estilo.css

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

body{
  font-family:arial,sans-serif;
  font-size:100%;
  margin:3em;
  background:#666;
  color:#fff;
}

h2,p{
  font-size:100%;
  font-weight:normal;
}
```

```

ul,li{
  list-style:none;
}

ul{ overflow:hidden;
padding:3em;
}

ul li a{
  text-decoration:none;
  color:#000;
  background:#ffc;
  display:block;
  height:10em; width:10em;
  padding:1em;
  -moz-box-shadow:5px5px7pxrgba(33,33,33,1);           /*Firefox*/
  -webkit-box-shadow:5px5px7pxrgba(33,33,33,.7);       /*Safari+Chrome*/
  box-shadow:5px5px7pxrgba(33,33,33,.7);              /*Opera*/
}

ul li{
  margin:1em;
  float:left;
}

ul li h2{
  font-size:140%; font-
weight:bold; padding-
bottom:10px;
}

ul li p{
  font-family:"Reenie Beanie",arial,sans-serif;
  font-size:180%;
}

ul li a{
  -webkit-transform:rotate(-6deg);
  -o-transform:rotate(-6deg);
  -moz-transform:rotate(-6deg);
}

ul li:nth-child(even) a{
  -o-transform:rotate(4deg);
  -webkit-transform:rotate(4deg);
  -moz-transform:rotate(4deg);
  position:relative;
  top:5px;
}

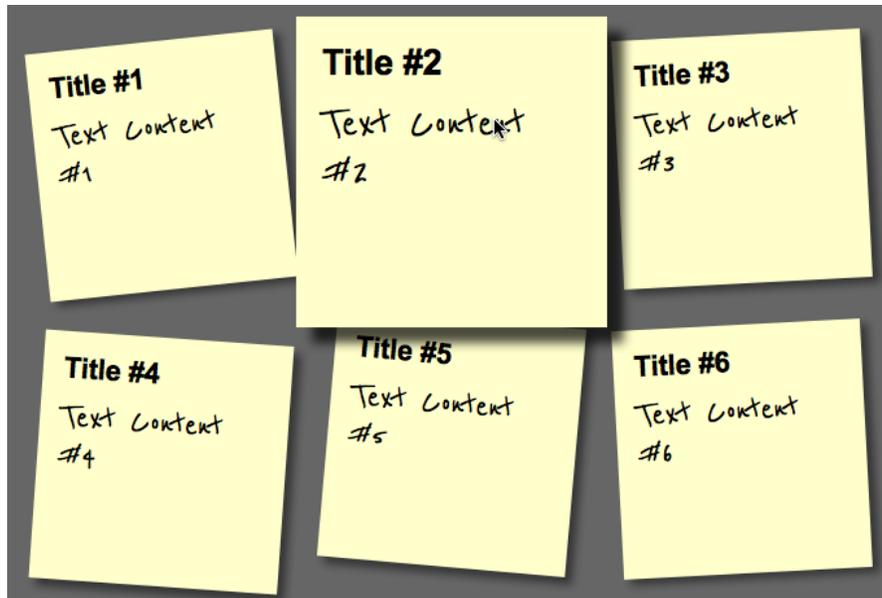
ul li:nth-child(3n) a{
  -o-transform:rotate(-3deg);
}

```

```
-webkit-transform:rotate(-3deg);  
-moz-transform:rotate(-3deg);  
position:relative;  
top:-5px;  
}
```

```
ul li:nth-child(5n) a{  
-o-transform:rotate(5deg);  
-webkit-transform:rotate(5deg);  
-moz-transform:rotate(5deg);  
position:relative;  
top:-10px;  
}
```

El paso 3 debe generar la siguiente pantalla:



Paso 4: Acercamiento al pasar el mouse sobre los stickies (hover)

Nuevamente estamos usando una funcionalidad de CSS3: transformación por escalamiento. Este tipo de transformación nos servirá para implementar un efecto de acercamiento (zooming) sobre el "sticky note".

El código fuente del archivo **estilo.css** se muestra a continuación:

```
estilo.c  
ss
```

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

```

}

body{
  font-family:arial,sans-serif;
  font-size:100%;
  margin:3em;
  background:#666;
  color:#fff;
}

h2,p{
  font-size:100%;
  font-weight:normal;
}

ul,li{
  list-style:none;
}

ul{ overflow:hidden;
  padding:3em;
}

ul li a{
  text-decoration:none;
  color:#000;
  background:#ffc;
  display:block;
  height:10em; width:10em;
  padding:1em;
  -moz-box-shadow:5px5px7pxrgba(33,33,33,1);           /*Firefox*/
  -webkit-box-shadow:5px5px7pxrgba(33,33,33,.7);      /*Safari+Chrome*/
  box-shadow:5px5px7pxrgba(33,33,33,.7);              /*Opera*/
}

ul li{
  margin:1em;
  float:left;
}

ul li h2{
  font-size:140%; font-
weight:bold; padding-
bottom:10px;
}

ul li p{
  font-family:"Reenie Beanie",arial,sans-serif;
  font-size:180%;
}

ul li a{
  -webkit-transform:rotate(-6deg);

```

```
-o-transform:rotate(-6deg);  
-moz-transform:rotate(-6deg);  
}
```

```
ul li:nth-child(even) a{  
-o-transform:rotate(4deg);  
-webkit-transform:rotate(4deg);  
-moz-transform:rotate(4deg);  
position:relative;  
top:5px;  
}
```

```
ul li:nth-child(3n) a{  
-o-transform:rotate(-3deg);  
-webkit-transform:rotate(-3deg);  
-moz-transform:rotate(-3deg);  
position:relative;  
top:-5px;  
}
```

```
ul li:nth-child(5n) a{  
-o-transform:rotate(5deg);  
-webkit-transform:rotate(5deg);  
-moz-transform:rotate(5deg);  
position:relative;  
top:-10px;  
}
```

```
ul li a:hover,ul li a:focus{  
-moz-box-shadow:10px 10px 7px rgba(0,0,0,.7);  
-webkit-box-shadow: 10px 10px 7px rgba(0,0,0,.7);  
box-shadow:10px 10px 7px rgba(0,0,0,.7);  
-webkit-transform: scale(1.25);  
-moz-transform: scale(1.25);  
-o-transform: scale(1.25);  
position:relative;  
z-index:5;  
}
```

Este paso debería producir un html similar al siguiente:

Title #1

Text Content
#1

Title #2

Text Content
#2

Title #3

Text Content
#3

Title #4

Text Content
#4

Title #5

Text Content
#5

Title #6

Text Content
#6