




**ACCIÓN FORMATIVA:**  
**< CSS BÁSICO >**

**BBVA**

## ICONO TRAINING



 [www.iconotc.com](http://www.iconotc.com)

 [linkedin.com/compan  
y/icono-training-  
consulting](https://www.linkedin.com/company/icono-training-consulting)

 [training@iconotc.com](mailto:training@iconotc.com)

## FORMADORA



Ana Isabel Vegas

Consultora / Formadora en Tecnologías de la Información.

¡Síguenos en las Redes Sociales!



# CSS BÁSICO










## DURACIÓN

 8 horas

## LUGAR/FECHAS /HORARIO

 Madrid. Día 11 de Abril de 2019. De 9:00 hs a 14:00 hs y 15:00 hs A 18:00 hs

## CONTENIDO:

-  Arquitectura, propósito y alcance de CSS
-  Inclusión de elementos CSS en páginas HTML5
-  Hojas de estilo, reglas y propiedades
-  Herencia de estilos: resolución de conflictos
-  Tratamiento de textos
-  Gestión de imágenes
-  Organización de tablas
-  Definición de estructura de una página HTML5
-  Transformaciones, transiciones y animaciones

## 1. CSS

### INTRODUCCIÓN A LAS HOJAS DE ESTILO

CSS es una tecnología para controlar las propiedades visuales de los Elementos que componen una página web. HTML se encarga de la estructuración de los contenidos y CSS se preocupa de la apariencia y del estilo siendo un complemento para hacer más atractivo el contenido de las páginas.

Con CSS se especifica cómo mostrar los elementos de una página, cómo será su presentación y su visualización.

CSS modifica propiedades como el posicionamiento, el tamaño, la fuente, el color, etc.,... de una forma precisa y exacta.

En 1996 el W3C crea CSS1, la primera versión de CSS, recoge propiedades para modificar el texto de un página, colores, fondos y bordes.

#### Para qué sirven las hojas de estilo

Con el uso de hojas de estilo el desarrollador de HTML puede generar una hoja de estilo patrón para todas las páginas de un sitio web, con el consiguiente ahorro de tiempo en diseño y sobretodo en mantenimiento y actualización del sitio.

Las hojas de estilo son complementos al HTML. Su misión es definir la apariencia y el estilo de los elementos. Por ejemplo, permite que especifiquemos en una línea que el texto del documento use la fuente Arial en color azul y con un tamaño de 12 puntos.

#### En HTML tendríamos que utilizar el siguiente código:

```
<b><font face="Arial" size="+2" color="blue">Texto en negrita</font></b>
```

#### En CCS:

```
<style type="txt/css">b{font-size:12; color:blue; font-family:Arial}</style>
<b>Texto en negrita</b>
```

### SINTAXIS

Las hojas de estilo tienen una estructura simple, flexible y potente.

Existen diferentes formas de definir los estilos dentro de un documento dependiendo de la finalidad a conseguir.

#### Declarando estilos en el documento HTML

Declarar estilos e introducirlos en una página web se puede realizar de tres formas:

- Usando la etiqueta HTML **<STYLE>**
- Usando el atributo HTML **"STYLE"**
- Usando una hoja de estilo externa.

### LA ETIQUETA HTML <STYLE>

Se separan los estilos del contenido de la página. Se pueden aplicar a más de un elemento.

```
<STYLE TYPE="text/css"> ;va dentro de la cabecera
    selector{nombrePropiedad: valor ;...}
</STYLE>
```

Al código selector{propiedad: valor} se le denomina regla.

La regla está formada por dos partes: selector y declaración.

El selector indica a qué elemento del documento se aplicará el estilo.

Dependiendo del tipo de selector hay distintas sintaxis.

La declaración es el código que va entre las llaves y que definen el estilo. A su vez, está formada por dos elementos la propiedad y el valor para dicha propiedad.

### EL ATRIBUTO <STYLE>

Permite definir estilos específicos para un solo elemento HTML. Se declara el estilo directamente en el elemento HTML usando el atributo STYLE.

```
<P STYLE ="font-family:Arial"; font-size:10px; color:#FFFFFF;"background-color:#000000">
```

**Párrafo formateado con CSS**

```
</P>
```

### HOJAS DE ESTILO EXTERNAS

Es el método más utilizado, se ponen las reglas en un fichero externo al texto con extensión **css** y se enlaza a dicho fichero desde la cabecera del documento HTML utilizando la etiqueta <LINK>.

```
<LINK REL="stylesheet" HREF=miestilo.css" TYPE="text/css">
```

## SELECTORES

Especifican a qué elementos van asignados los estilos creados.

Pueden ser de dos tipos:

- I. **SELECTORES SIMPLES**
- II. **SELECTORES CONTEXTUALES**

### I. **SELECTORES SIMPLES**

Identifican los elementos HTML a los que se asignará el estilo según el Tipo de elemento que sea o un atributo que posea.

**TD{color:blue} Todas las celdas de la página irán en azul**

Existen varios tipos de selectores simples:

1. **Selectores de tipos**
2. **Selectores de clases**
3. **Selectores de ID**
4. **Selector universal**

## 1. Selectores de tipos

El estilo se asigna a todos los elementos del tipo que se indique en la Regla.  
BODY {font-family: Arial}

## 2. Selectores de clases

Son los más utilizados, se asigna un estilo a cualquier elemento HTML utilizando el atributo CLASS.  
sintaxis : .nombreEstilo {propiedad: valor}  
<ETIQUETA class="nombreEstilo">TEXT</ETIQUETA>

## 3. Selectores de ID

Se basan en el atributo **ID** de los elementos **HTML** y aplican el estilo a cualquier elemento que tenga ese identificador, luego sólo sirve para un elemento pues el valor de id tiene que ser único.  
sintaxis : #nombreEstilo {propiedad: valor}  
<ETIQUETA id="nombreEstilo">TEXT</ETIQUETA>

## 4. Selector universal

Permite asignar un estilo a todos los elementos de la página  
\*{propiedad: valor}

## II. SELECTORES CONTEXTUALES

Se utilizan menos porque no son compatibles al cien por cien entre los navegadores.  
Se aplica el estilo según la posición que ocupa el elemento dentro de la estructura de la página.  
sintaxis : ETIQUETA1 ETIQUETA2 {propiedad: valor} Todas  
ETIQUETA2 que esté dentro de ETIQUETA1 tendrá valor en propiedad

## PSEUDO ELEMENTOS

Permiten acceder a elementos específicos que no son identificables en HTML. Son cuatro:  
:first-line: Se refiere a la primera línea del elemento especificado.  
:first-letter: Permite acceder a la primera letra del elemento.  
:before: Permite acceder al elemento anterior al otro.  
:after: Permite acceder al elemento posterior al otro.  
Ejemplos: TD:first-line {color:navy}P:first-letter{font-size: 16}

## PSEUDO CLASES

Hacen referencia a elementos como enlaces o el primer hijo de cada Elemento. Son:  
:first-child: Hace referencia al primer hijo de otro elemento  
:link: Hace referencia a los enlaces sin visitar  
:visited: Hace referencia a los enlaces visitados  
:hover: Permite modificar el estilo del enlace sobre el que se pasa el Ratón.  
Ejemplos: A:link{color:blue}A:hover{text-decoration:none}

## HERENCIA

En CSS los hijos, la mayoría de las veces heredan los estilos de sus Padres siempre que el hijo no tenga ningún estilo aplicado.

No todas las propiedades se heredan, pero se puede utilizar la palabra "inherit" como valor de una propiedad para asegurarnos que se Heredará.

## CASCADA

Las propiedades de los elementos situados más arriba en la jerarquía de la página ceden las propiedades a sus hijo. Esto da lugar a dos problemas:

1. A veces, un elemento hijo hereda propiedades no deseadas.
2. Si en una página hay múltiples estilo resulta difícil determinar qué elemento obtiene la propiedad.

Para solucionarlo, se utiliza el atributo **!important** que indica que éste estilo tiene preferencia sobre cualquier otro.

**Ejemplo:** BODY{font-family:Ariel}  
P{font-family:Times !Important}

## ETIQUETAS DIV Y SPAN

Permiten estructurar los documentos HTML.

DIV define un bloque de información.(salto de línea)

SPAN define el contenido de un elemento dentro del bloque

## PRINCIPALES PROPIEDADES CSS

### COMENTARIOS /\* \*/

### UNIDADES DE MEDIDA Y COLOR

Se forman con el signo + ó –seguido de un número, de una Abreviatura de dos letras que indica la unidad.

Las unidades de longitud pueden ser relativas o absolutas:

- **Unidades relativas son:**
  1. **em:** un em es igual al tamaño de la fuente del elemento actual
  2. **ex:** un ex es x-altura de un conjunto de caracteres
  3. **px:** píxeles
- **Unidades absolutas son:**
  1. **in:** pulgadas
  2. **cm:** centímetros
  3. **mm:** milímetros
  4. **pt:** puntos(1pt=1/72 pulgada)
  5. **pc:** picas(1pc=12pt)

También se pueden establecer valores de porcentaje mediante % para definir colores se usa la palabra **color** en especificación

**RGB#rrggbb rgb(n,n,n) rgb(%,%,%)**

### PROPIEDADES PARA EL TEXTO

- **font-family**: fuente de texto
- **font-size**: tamaño de fuente
- **font-style**: estilo de fuente (normal, italic (cursiva), oblique (inclinada))
- **font-weight**: constants (bold, bolder, lighter, normal)

### PROPIEDADES DE COLOR Y FONDO

- **color**
- **background,background-color**: color de fondo
- **background-image**: imagen de fondo de un elemento

### PROPIEDADES DE BORDE Y MARGEN

- **border-width**
- **border-color**
- **border-style**
- **margin** especifica margen
- **padding** distancia entre el borde y contenido

### PROPIEDADES DEL PUNTERO DEL RATON

- **cursor** controla el puntero

### BARRAS DE DESPLAZAMIENTO

- **scrollbar-base-color**
- **scrollbar-face-color**
- **scrollbar-track-color**
- **scrollbar-arrow-color**
- **scrollbar-highlight-color**
- **scrollbar-shadow-color**
- **scrollbar-darkshadow-color**

## OBJETIVOS

- Separar la estructura del documento (**etiquetas HTML**) del estilo de presentación (**hojas de estilo**)
- Estándar del W3C: CSS (Cascade Style Sheets)

Versión actual:**CS33**, pero...

- El soporte en los navegadores actuales es muy pobre
- CSS2:



- Explorer 6 , Navegador 7 ,Mozilla y Firefox lo implementan aceptablemente
- Las versiones anteriores, muy parcialmente

## REGLAS DE ESTILO

- Asocia una **etiqueta HTML** con su **aspecto**

`h1 {color:blue; font-size:25px;}`

- El aspecto se “hereda”, considerando el HTML como un “árbol genealógico”

`body {color:blue; font-size:25px;}`

`p {font-size:15px;}`

¿Dónde colocar las reglas de estilo?

- En un fichero aparte (.css)

```
<head>
  <link href="estilo.css" rel="stylesheet" type="text/css" media="screen">
  ...
</head>
```

- En la cabecera

```
<head>
  <style type="text/css" media="screen">
    h1 {color:red}
  </style>
</head>
```

- En el atributo style de una etiqueta
  - Cuando el estilo sólo se utilice una vez
  - Sólo se especifican las propiedades

`<h1 style="color:blue; font-size:3em"> Hola, amigos </h1>`

- Etiquetas:**

- Se pueden poner varias etiquetas en la misma regla

`body, td {background:blue}`

- Que afecte a una etiqueta sólo si está dentro de otra. Afecta a la de la derecha

```
h1 em {color:red}
<h1> <em> Yo soy rojo </em> </h1> <em> pero yo no </em>
```

- ¿Cómo cambiar el estilo sólo de cierto/s h1 y no de todos?

- Se pueden poner varias etiquetas en la misma regla

- Clase** (para utilizar varias veces)

```
h1.azul {color:blue; font-size:25px;} /* solo vale para h1 */
.hortera {color:red; background:yellow;} /* para cualquier etiqueta */
...
<h1 class="azul"> Un título azul </h1>
<p class="hortera"> cómo molo, eh <p>
```

- **Identificador(sólo una vez)**

```
h1#azul {color:blue; font-size:25px;} /* solo vale para h1 */
#hortera {color:red; background:yellow;} /* para cualquier etiqueta */
```

```
<h1 id="azul"> EL título azul </h1>
```

- ¿Cómo cambiar de estilo cierta sección del texto para la que no tenemos etiqueta HTML?

```
<h1>Bienvenidos a mi web </h1>
```

- Utilizar las etiquetas HTML **<div>** y **<span>** que sirven simplemente para definir secciones de texto.

- Diferencia: <div> fuerza retorno de carro antes y después <span> no

```
<h1>Bienvenidos a <span class="rojo">mi web </span> </h1>
```

- Pseudoclases: estados de etiquetas

clase a vinculada a <A>

- a:visited Enlace ya visitado
- a:hover Enlace por debajo del ratón

```
a:hover {background:yellow}
```

- Pseudoelementos: partes de elementos

- :first-letter
- :first-line

```
p:first-letter {font-size:3em}
```

## TIPOS DE PROPIEDADES

- **FUENTES:** tipos de letras, tamaño, negrita, cursiva...
- **COLOR:** de texto, de fondo, imagen de fondo...
- **TEXTO:** justificación, espacio entre letras y/o palabras.
- **CAJA:** bordes a los 4 lados
- **FORMATO VISUAL:** posición en la página, mostrar o no..
- **TABLAS**
- **LISTAS, NUMERACIÓN AUTOMÁTICA**

### Unidades de Medida

- **Palabras clave:** small, border
- **Unidades absolutas:** px(pixeles), cm ,mm , in(pulgadas), pt(=1/72in), pc(=12pt)
- **Unidades relativas:** em(=tamaño por defecto).EJ: 2em=2veces el tamaño por defecto.
- **URLs:** urls
- **COLORES:** yellow,#ff0000,rgb(255,0,0),rgb(100%,0%,0%)

### Propiedades de fuentes

```
p {
    font-size:25px;                /* tamaño */
    font-family: "Arial narrow", sans-serif; /* fuente */
    font-weight: 400                /* negrita */
}
```

### Color

```
p {
    color:rgb(0,0,255);            /* color texto */
    background-image: url("mosca.gif"); /* imagen fondo */
    background-repeat: repeat       /* fondo mosaico */
}
```

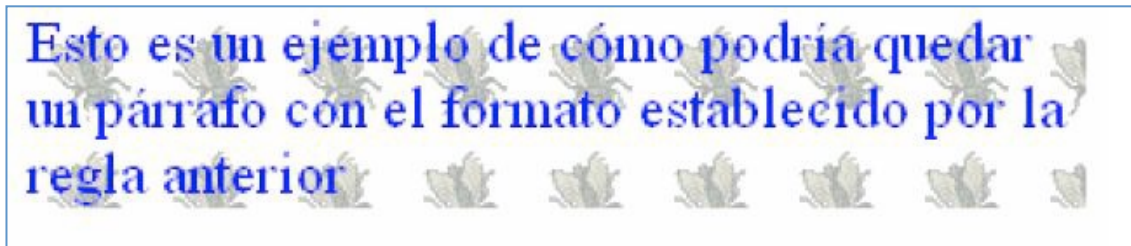


Gráfico . Ejemplo CSS texto con imagen de fondo

### Texto

```
p {
    letter-spacing:0.5em;           /* espaciado letras */
    text-indent: 3em;              /* indentación primera línea */
}
```

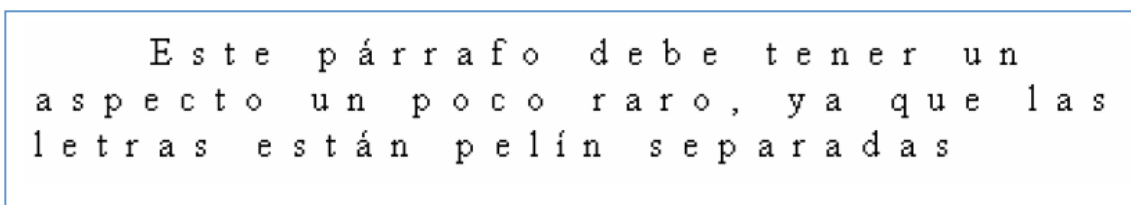


Gráfico. Ejemplo CSS de texto

## Caja

```
#aviso {
padding-left: 50%;
padding-right: 20px;
padding-top: 3em;
padding-bottom: 1em;
background-color: #FFEFD5;
border: 1px solid;
}
```

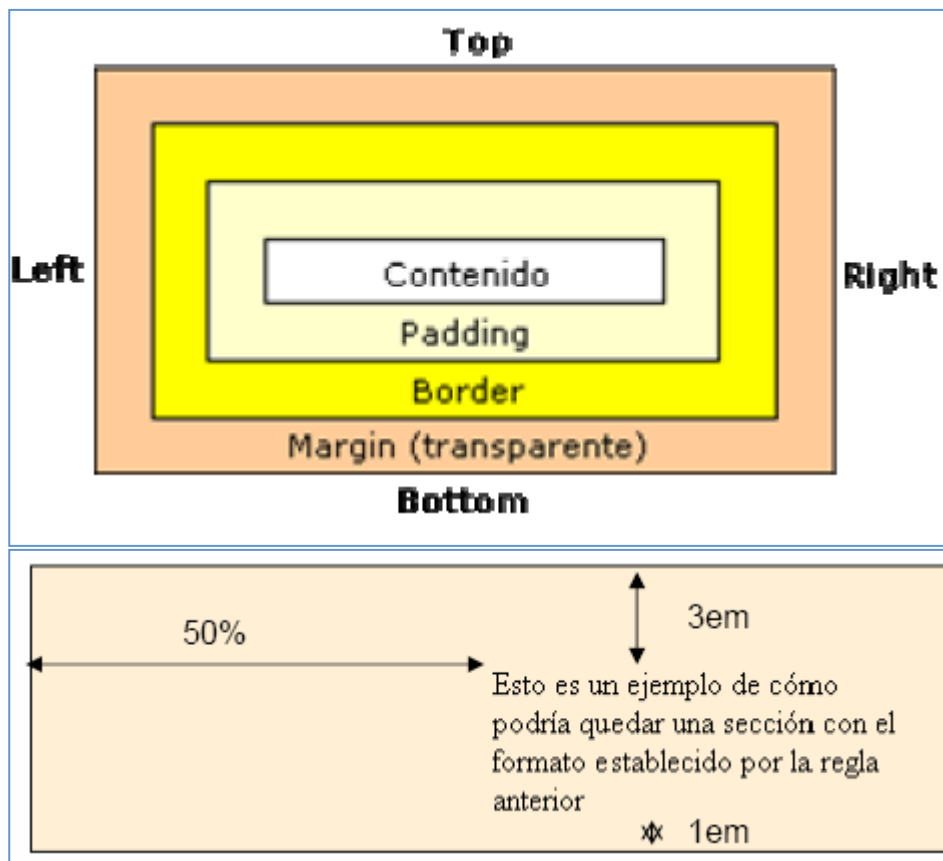


Gráfico. Ejemplo CSS distribución del texto

## Dimensiones de caja

- Para especificar el ancho y alto : propiedades **width** y **height**
- El Explorer incumple el estándar
  - Explorer: **width=contenido+padding+border**
  - Estándar: **width = sólo contenido**

```
#aviso {  
    width:100px;  
    background-color : Yellow;  
    padding: 50px 25px 50px 25px;  
    border: 1px solid;  
}
```

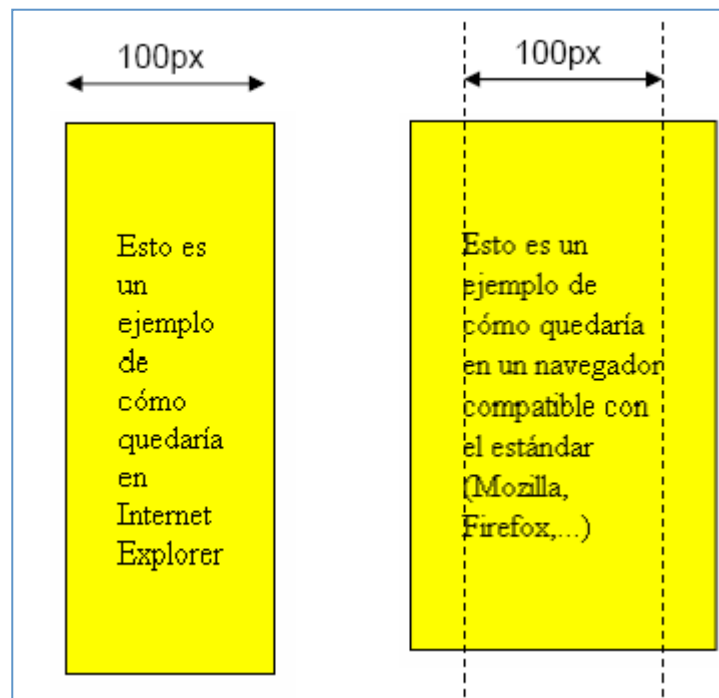


Gráfico. Ejemplo CSS ancho de la caja de texto

### Formato visual: Posicionamiento

- Permiten especificar el tipo de posicionamiento: atributo **position**
  - **absolute**: referente al elemento contenedor del posicionado. Es como si se sacara de la página y se pusiera en otra capa.
  - **fixed**: referente a la ventana(no funciona en el explorer)
  - **relative**: referente a la posición que ocuparía normalmente. Deja un hueco en dicha posición
- Permiten especificar posición
  - **left**: x, **top**: y(origen de coords en la esquina superior izquierda)
  - **z-index**: z(cuanto mayor .más cerca)

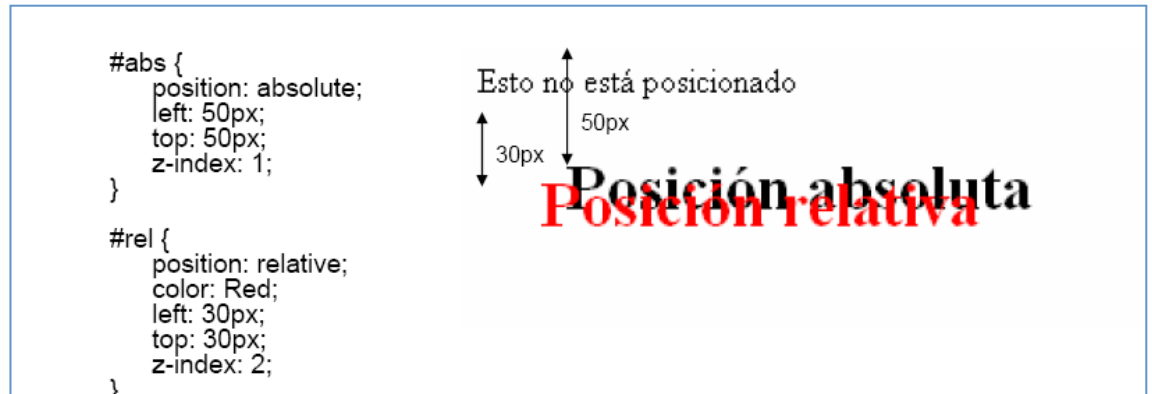


Gráfico. Posición relativa y absoluta con CSS

### La Cascada CSS

- Los efectos CSS se aplican “**en cascada**”. El color puede venir de un estilo en un .css mientras que el tamaño viene de un estilo en la cabecera.
- Si los estilos son contradictorios, se emplean varios criterios.
  - Quién ha definido el estilo(autor, usuario, navegador): normalmente tiene prioridad el autor, salvo **!important**  
     body {font-size: 18px !important}
  - Especificidad: ganan los selectores más específicos
  - Orden de definición: gana la última regla

## 2. CSS3

### BORDES REDONDEADOS (BORDER-RADIUS)

La propiedad border-radius permite crear esquinas redondeadas. Especificamos en píxeles u otra medida el radio de redondeo de la o las esquinas.

Podemos indicar un único valor que se asignará a los cuatro vértices:

```
#recuadro1{  
  border-radius: 20px;  
  background-color:#ddd;  
  width:200px;  
  padding:10px;  
}
```

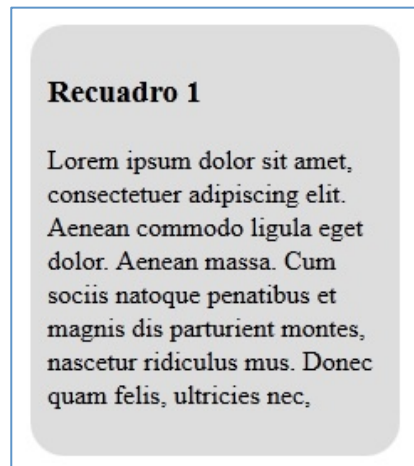


Gráfico. Ejemplo bordes redondeados

También podemos indicar el redondeo de cada vértice en forma independiente (el orden de los valores son la esquina superior izquierda, la esquina superior derecha, la esquina inferior derecha y por último la esquina inferior izquierda):

```
#recuadro2{  
  border-radius: 20px 40px 60px 80px;  
  background-color:#aa0;  
  width:200px;  
  padding:10px;  
  margin-top:10px;  
}
```

### Recuadro 2

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.  
Aenean commodo ligula eget  
dolor. Aenean massa. Cum  
sociis natoque penatibus et  
magnis dis parturient montes,  
nascetur ridiculus mus. Donec  
quam felis, ultricies nec,

Gráfico . Bordes redondeados desiguales

## SOMBRAS (BOX-SHADOW)

La propiedad box-shadow permite definir una sombra a un objeto de la página. Debemos definir tres valores y un color, por ejemplo:

```
#recuadro1{  
  box-shadow: 30px 10px 20px #aaa;  
  border-radius: 20px;  
  background-color:#ddd;  
  width:200px;  
  padding:10px;  
}
```

### Recuadro 1

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.  
Aenean commodo ligula eget  
dolor. Aenean massa. Cum  
sociis natoque penatibus et  
magnis dis parturient montes,  
nascetur ridiculus mus. Donec  
quam felis, ultricies nec,

Gráfico. Aplicar sombras

La propiedad tiene 3 valores y un color, los valores son los siguientes:



- El desplazamiento horizontal de la sombra, positivo significa que la sombra se encuentra a la derecha del objeto, un desplazamiento negativo pondrá la sombra a la izquierda.
- El desplazamiento vertical, uno negativo la sombra será en la parte superior del objeto, uno positivo la sombra estará por debajo.
- El tercer parámetro es el radio de desenfoque, si se pone a 0 la sombra será fuerte y con color liso, más grande el número, más borrosa será.

El último valor es el color a aplicar a la sombra.

## SOMBRAS DE TEXTO (TEXT-SHADOW)

Es importante tener en cuenta que esta propiedad no la soporta el Internet Explorer 9.  
La propiedad text-shadow nos permite definir una sombra a un texto, la sintaxis más común es:

```
Elemento {  
    text-shadow: desplazamientoX desplazamientoY radio-de-desenfoque color;  
}
```

La propiedad tiene 4 valores que son los siguientes:

- El desplazamiento horizontal de la sombra, un desplazamiento negativo pondrá la sombra a la izquierda.
- El desplazamiento vertical, un valor negativo dispone la sombra en la parte superior del texto, uno positivo la sombra estará por debajo del texto.
- El tercer parámetro es el radio de desenfoque, si se pone a 0 la sombra será fuerte y con color liso, más grande el número, más borrosa será.
- El último parámetro es el color de la sombra.

Por ejemplo si queremos que un texto tenga una sombra en la parte inferior a derecha con un pequeño desenfoque de color gris luego debemos implementar el siguiente código:

```
#titulo1 {  
    text-shadow: 5px 5px 5px #aaa;  
}
```



Gráfico. Texto con sombra inferior

Si queremos que la sombra se disponga en la parte superior izquierda de cada letra luego debemos definir los siguientes valores:

```
#titulo2 {  
  text-shadow: -5px -5px 5px #aaa;  
}
```



Gráfico. Texto con sombra superior

Otra sintaxis de text-shadow es aplicar varias sombras al texto, por ejemplo:

```
#titulo3 {  
  text-shadow: 3px 3px 5px #f00,  
              6px 6px 5px #0f0,  
              9px 9px 5px #00f;  
}
```



Gráfico. Texto con múltiples sombras

## TRANSFORMACIONES 2D: ROTACIÓN (TRANSFORM: ROTATE)

Las transformaciones todavía no están definidas como un estándar en todos los navegadores, por lo que es necesario agregar el prefijo del navegador que la implementa:

```
Elemento {  
  -ms-transform: función de transformación(valor(es)); /* Internet Explorer */  
  -webkit-transform: función de transformación(valor(es)); /* WebKit */  
  -moz-transform: función de transformación(valor(es)); /* Firefox */  
  -o-transform: función de transformación(valor(es)); /* Opera */  
}
```

Tengamos en cuenta que la propiedad de transformación 2D definitiva será:

```
Elemento {  
  transform: función de transformación(valor(es));  
}
```

La primer función de transformación que veremos será la de rotar un elemento HTML.

La función de rotación se llama rotate y tiene un parámetro que indica la cantidad de grados a rotar.

La rotación es en el sentido de las agujas del reloj. Podemos indicar un valor negativo para rotar en sentido antihorario.

Para rotar un recuadro 45 grados en sentidos de las agujas de un reloj y que funcione en la mayoría de los navegadores deberemos implementar el siguiente código:

```
#recuadro1{  
-ms-transform: rotate(45deg);  
-webkit-transform: rotate(45deg);  
-moz-transform: rotate(45deg);  
-o-transform: rotate(45deg);  
transform: rotate(45deg);  
border-radius: 20px;  
background-color:#ddd;  
width:200px;  
padding:10px;  
}
```

Tengamos en cuenta que se ejecuta la propiedad -ms-transform, -webkit-transform etc. según el navegador que está procesando la página, inclusive hemos agregado la propiedad transform: rotate(45deg) que será la que en un futuro todos los navegadores interpretarán.

El resultado visual es el siguiente:

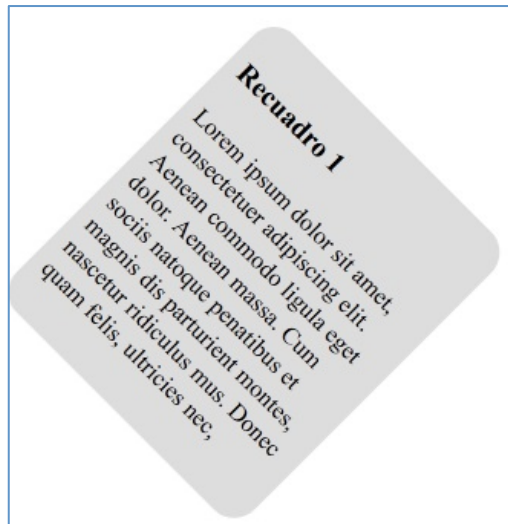


Gráfico. Rotación de texto

Otra cosa importante para notar es que el punto de rotación coincide con el centro del recuadro (es como clavar una alfiler en el centro del recuadro y luego rotar el recuadro en el sentido de las agujas del reloj).

También podemos rotar en sentido antihorario indicando el valor del grado con un valor negativo:

```
#recuadro2{  
-ms-transform: rotate(-45deg);  
-webkit-transform: rotate(-45deg);  
-moz-transform: rotate(-45deg);  
-o-transform: rotate(-45deg);  
transform: rotate(-45deg);  
border-radius: 20px;
```

```
background-color:#ddd;
width:200px;
padding:10px;
}
```

También podemos observar que cuando la rotación se ejecuta no ocupa más espacio el elemento HTML sino que se solapa eventualmente con otros elementos de la página:

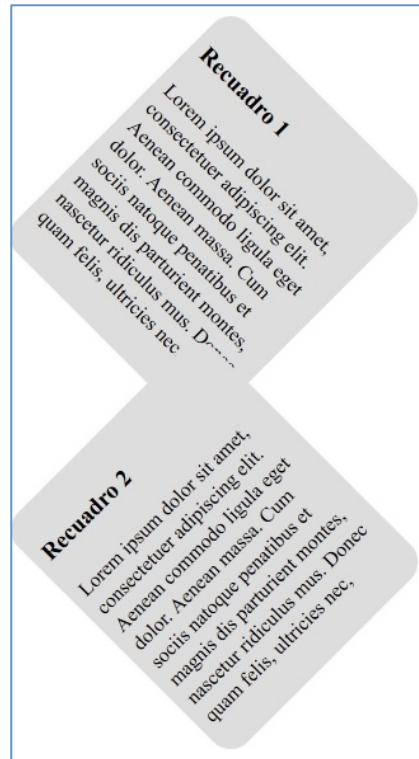


Gráfico. Solapamiento de elementos con rotación

## OPACIDAD (OPACITY)

La opacidad es una característica de los objetos de no dejar pasar la luz (mientras un objeto es más opaco significa que no deja pasar la luz) Un elemento HTML dispone de la propiedad opacity para definir cuál es su opacidad. La sintaxis es la siguiente:

```
Elemento {
  opacity: valor;
}
```

El valor es un número comprendido entre 0 y 1. El 0 significa que es totalmente transparente (luego no se verá nada en pantalla, pero el espacio ocupado por el elemento HTML queda reservado), el 1 significa que es totalmente opaco (no deja pasar la luz)

Veamos tres recuadro con una imagen de fondo y un texto en su interior con diferentes niveles de opacidad (tengamos en cuenta que cuando le asignamos una opacidad a un elemento HTML luego todos los elementos contenidos en dicho elementos heredan dicha opacidad):

```
#recuadro1 {  
  background-image: url("foto1.jpg");  
  opacity:0.3;  
  color:#f00;  
  width:700px;  
  height:450px;  
  border-radius:15px;  
  font-size:30px;
```

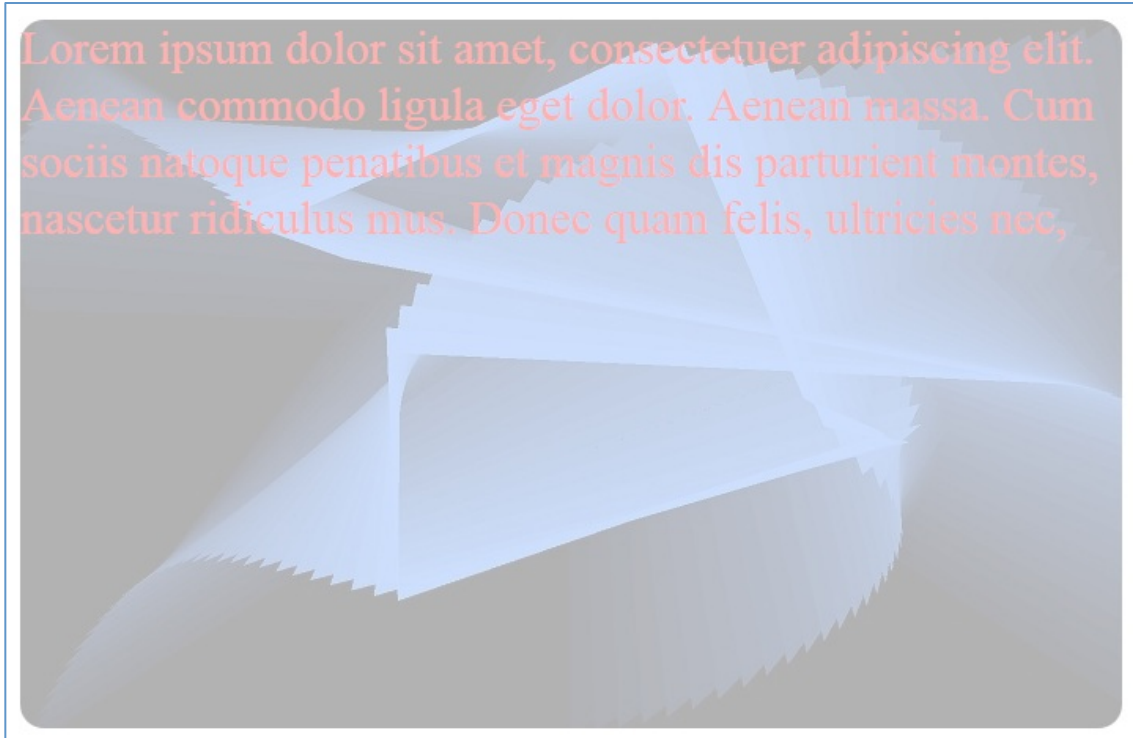


Gráfico. Ejemplo con poca opacidad

Ahora con una opacidad mayor:

```
#recuadro2 {  
  background-image: url("foto1.jpg");  
  opacity:0.6;  
  color:#f00;  
  width:700px;  
  height:450px;  
  border-radius:15px;  
  font-size:30px;  
}
```

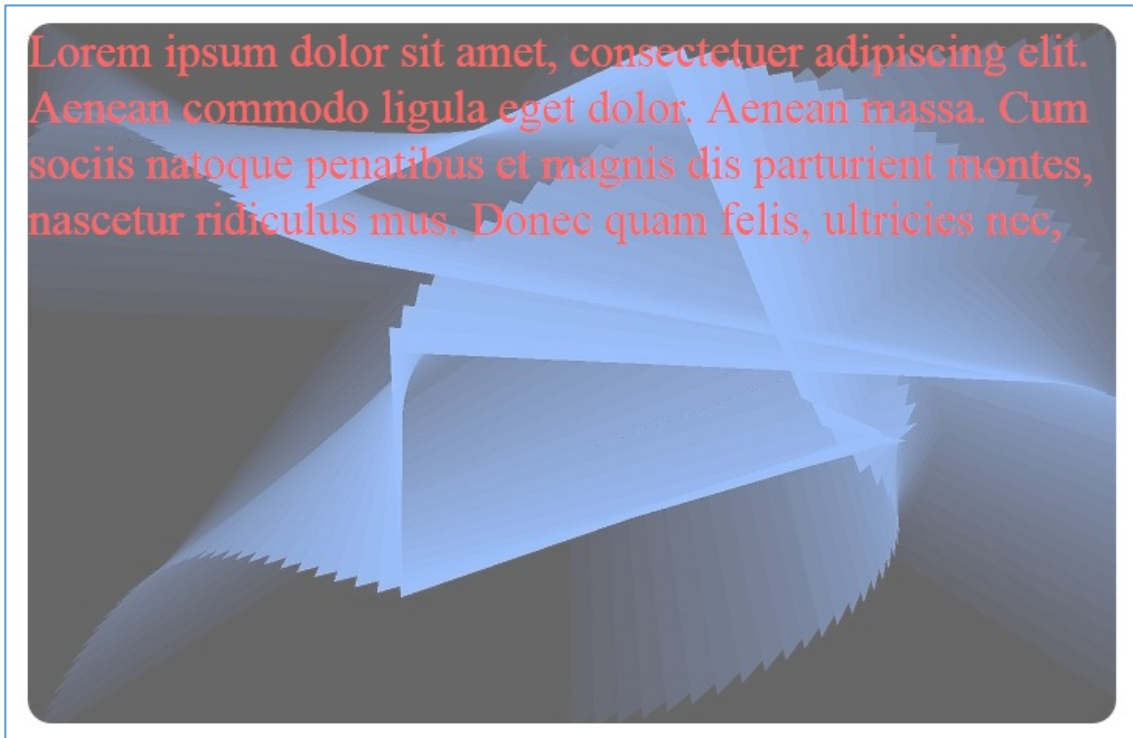


Gráfico. Elemento con poca opacidad

Finalmente con una opacidad de 0.9 (casi no deja pasar nada de luz):

```
#recuadro3 {  
  background-image: url("foto1.jpg");  
  opacity:0.9;  
  color:#f00;  
  width:700px;  
  height:450px;  
  border-radius:15px;  
  font-size:30px;  
}
```

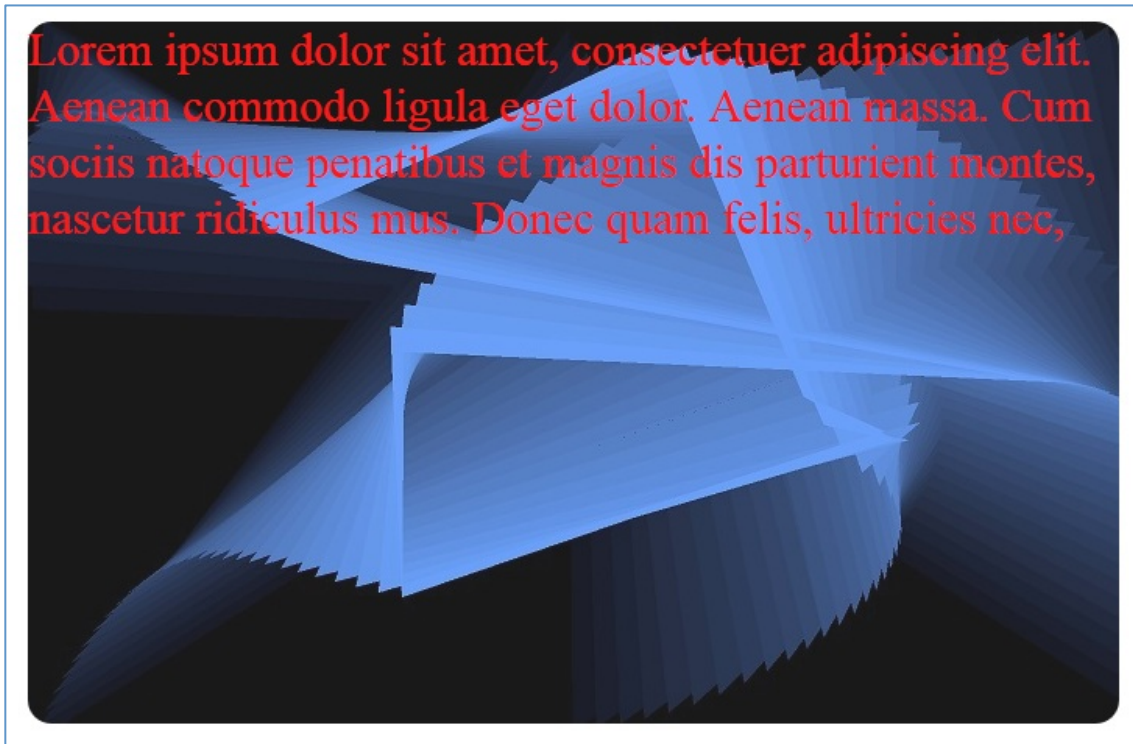


Gráfico. Elemento con mucha opacidad

Veamos qué pasa si no disponemos opacidad en el recuadro y el texto tiene un color de fondo:

```
#recuadro4 {
  background-image: url("foto1.jpg");
  color:#f00;
  width:700px;
  height:450px;
  border-radius:15px;
  font-size:30px;
}

.texto4 {
  background-color:yellow;
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,

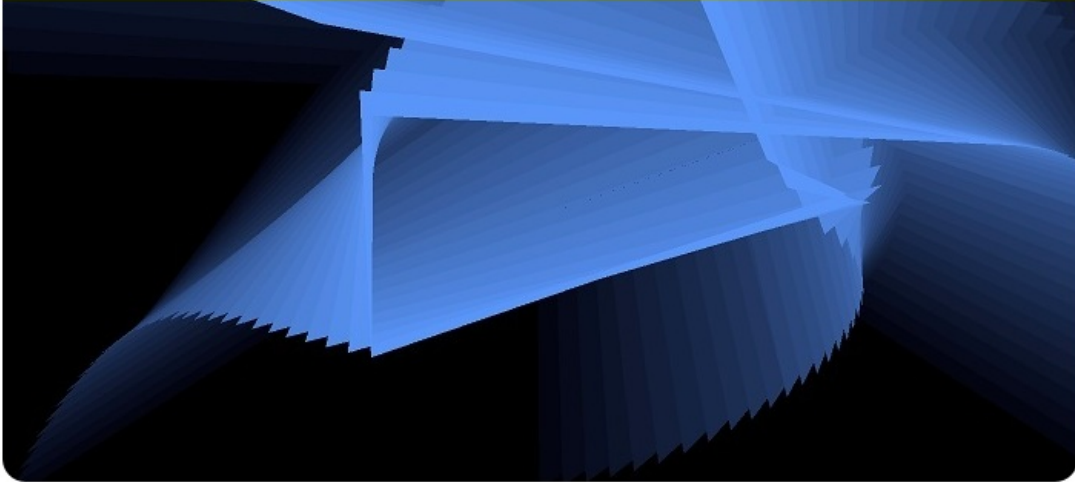


Gráfico. Elemento sin opacidad

Y luego definiendo una opacidad en el párrafo:

```
#recuadro5 {  
  background-image: url("foto1.jpg");  
  color:#f00;  
  width:700px;  
  height:450px;  
  border-radius:15px;  
  font-size:30px;  
}
```

```
.texto5 {  
  background-color:yellow;  
  opacity:0.2;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec,

Gráfico. Opacidad en un párrafo

## MÚLTIPLES COLUMNAS (COLUMN-COUNT)

Otra novedad que nos trae CSS3 es permitir disponer un bloque de texto en múltiples columnas con la única indicación de la cantidad de columnas que queremos que lo divida:

```
Elemento {  
    column-count: cantidad de columnas;  
}
```

Por ejemplo para generar un cuadro con tres columnas debemos implementar el siguiente código:

```
#recuadro1{  
    -moz-column-count:3;  
    -webkit-column-count:3;  
    column-count:3;  
    border-radius: 20px;  
    background-color:#ddd;  
    width:600px;  
    padding:10px;  
}
```

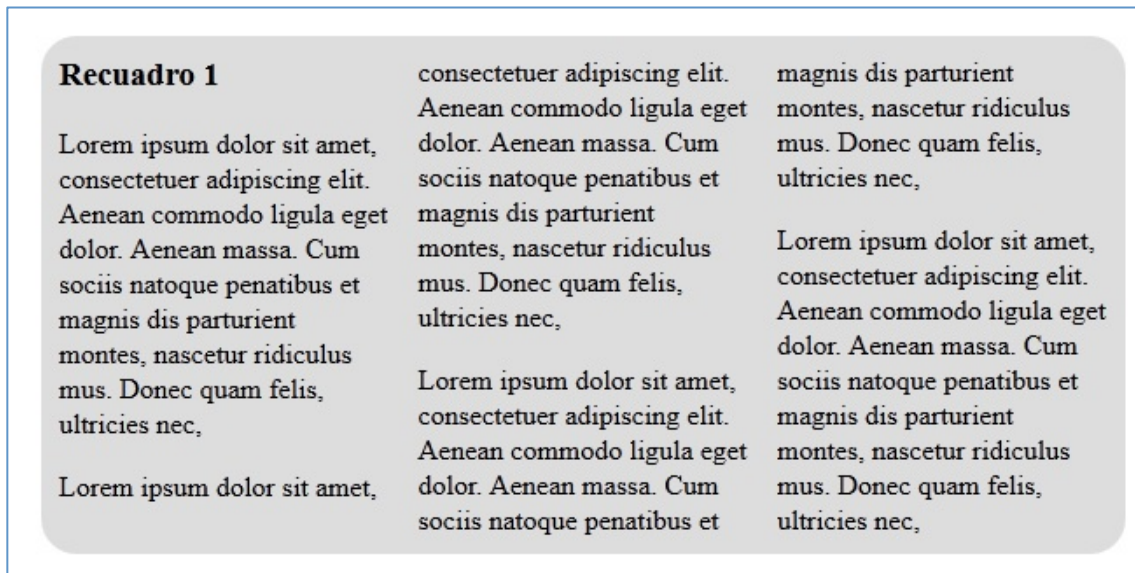


Gráfico . Texto en varias columnas

Tenemos que anteceder los prefijos -moz, -webkit para que funcionen con el Firefox y Chrome. El Opera ya implementa la propiedad definitiva column-count y hasta la versión 9 de IEExplorer no ha implementado el column-count.

## IMÁGENES DE FONDO (BACKGROUND-IMAGE)

La propiedad background-image permite insertar un conjunto de imágenes dentro de un elemento. Para ello debemos especificar sus nombres:

```
Elemento {
  background-image: url("imagen1", url("imagen2", ...));
}
```

La primera imagen que se dibuja es la que indicamos al final de la lista.

Por ejemplo vamos a mostrar una imagen de fondo con formato jpg y sobre esta una de tipo png, la primera:

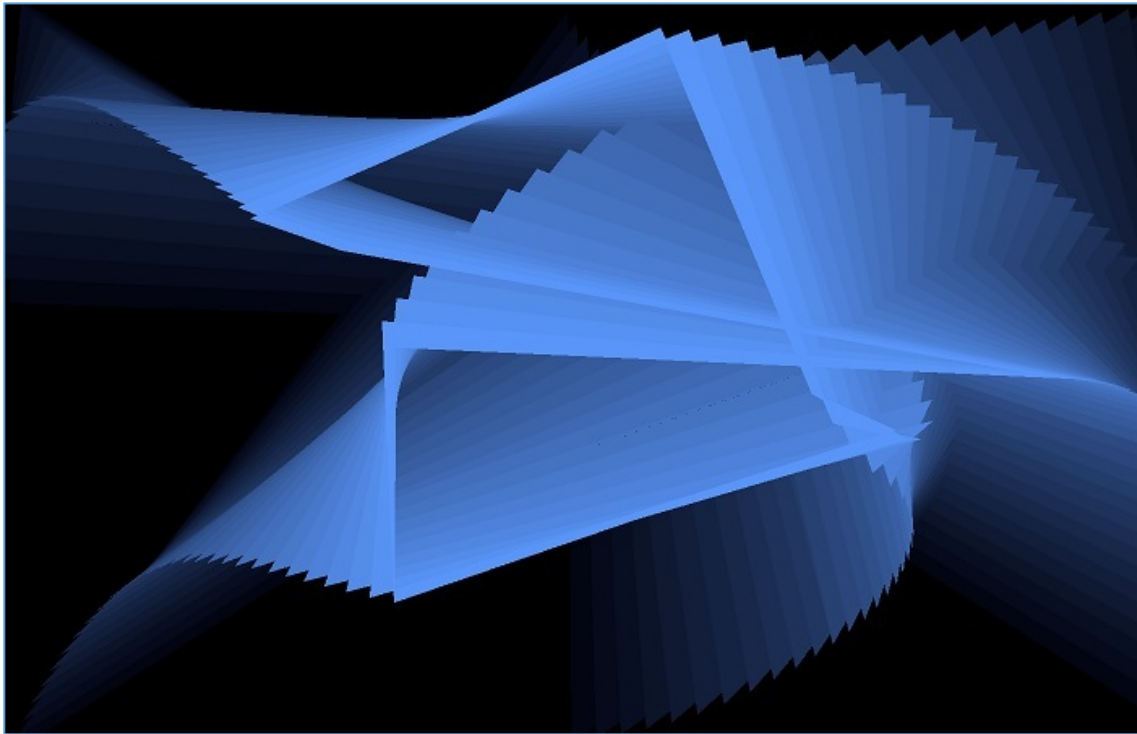


Gráfico. Imagen de fondo

y la imagen png es:

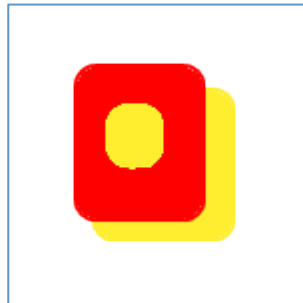


Gráfico. Imagen de muestra

Luego la página que muestra superpuestas las dos imágenes en un recuadro es:

```
<!DOCTYPE html>
<html>
<head>
<title>Prueba</title>

<style type="text/css">

#recuadro1{
  background-image: url("logo1.png"), url("foto1.jpg");
  background-repeat: no-repeat;
```

```
width:700px;  
height:450px;  
}  
  
body {  
  background:white;  
  margin:50px;  
}  
  
</style>  
  
</head>  
<body>  
  
<div id="recuadro1">  
</div>  
  
</body>  
</html>
```

El resultado es:

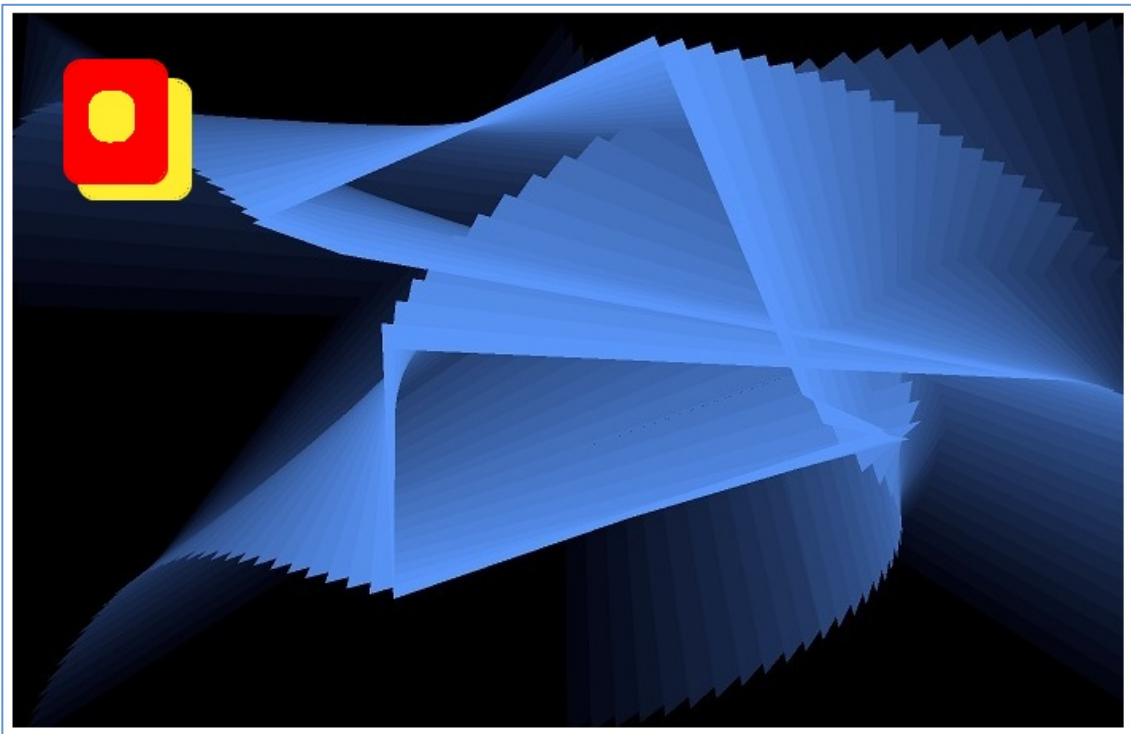


Gráfico. Fondo con dos imagenes

Para esto definimos:

```
#recuadro1{  
  background-image: url("logo1.png"), url("foto1.jpg");  
  background-repeat: no-repeat;  
  width:700px;  
  height:450px;
```

El recuadro coincide con el tamaño de la imagen "foto1.jpg" (700\*450) Luego la imagen "logo1.png" es de 150\*150 píxeles.

Como vemos primero se dibuja la imagen "foto1.jpg" (que es la última) y luego sobre esta la imagen "logo1.png". Otro cosa importante es inicializar la propiedad background-repeat con el valor no-repeat. En caso de no inicializar dicha propiedad tendremos a la imagen logo1.png repetida dentro del recuadro:

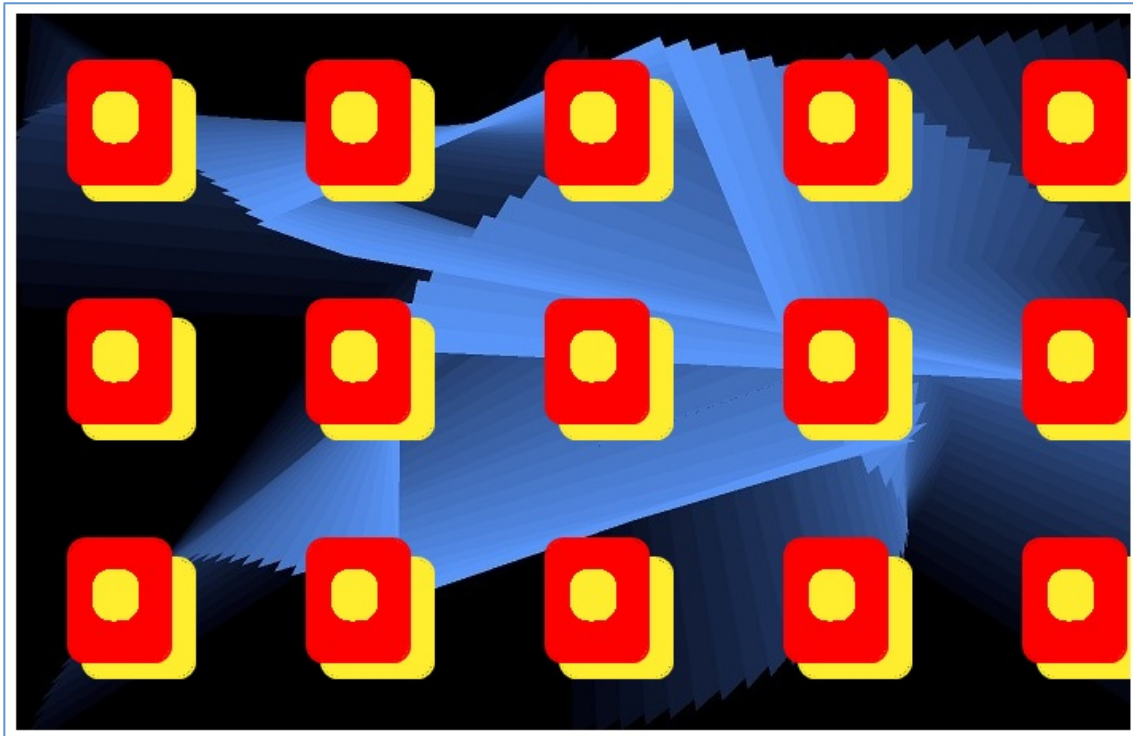


Gráfico. Repetición de imagen de fondo

## TRANSICIONES DE UNA PROPIEDAD (TRANSITION)

Las transiciones en css3 permiten modificar el valor de una propiedad de un elemento HTML en forma gradual durante un tiempo determinado de un estado inicial a un estado final.

La sintaxis más simple para definir una transición de una propiedad es:

```
Elemento {
  transition: [nombre de propiedad] [duración de la transición] ;
}
```

Por ejemplo crearemos dos recuadro y cuando pasemos la flecha del mouse sobre el mismo cambiaremos su tamaño. El primer recuadro lo haremos sin transición:

```
#recuadro1{
  border-radius: 20px;
  background-color:#ddd;
  width:200px;
  padding:10px;
}
```

```
#recuadro1:hover{  
  width:300px;  
}
```

Cuando probamos la página veremos que el recuadro1 cambia el ancho del recuadro de 200 píxeles a 300 píxeles en forma instantanea al pasar la flecha del mouse.

Ahora si definimos la propiedad transition indicando que actúe sobre la propiedad width y que el cambio lo realice en 1 segundo:

```
#recuadro2{  
  border-radius: 20px;  
  background-color:#ff0;  
  width:200px;  
  padding:10px;  
  transition:width 1s;  
  -moz-transition:width 1s;  
  -ms-transition:width 1s;  
  -webkit-transition:width 1s;  
  -o-transition:width 1s;  
}
```

```
#recuadro2:hover{  
  width:300px;  
}
```

Luego tenemos la siguiente sintaxis:

```
transition:width 1s;
```

Como vemos indicamos que la transición afecta a la propiedad width y que el cambio se realice en 1 segundo. Debemos agregar la propiedad con cada prefijo de navegador hasta que se estabilice su uso.

## ANIMACIONES (SINTAXIS BÁSICA)

Las animaciones en css3 nos permiten hacer cosas que con las transiciones no alcanzamos y nos quedamos cortos.

La sintaxis básica para una animación:

```
Elemento {  
  animation-name: [nombre de la animación];  
  animation-duration: [tiempo de duración];  
}
```

```
@ keyframes [nombre de la animación] {  
  from {  
    [propiedades y valores del estado inicial de la animación]  
  }  
  to {  
    [propiedades y valores del estado final de la animación]  
  }  
}
```

Veamos un ejemplo donde emplearemos la sintaxis expuesta, a este momento solo el FireFox, Chrome y Opera implementan las animaciones. Mostraremos un recuadro con un texto que parta de la columna 30px y avance hasta la columna 300px y que cambie del color gris al color amarillo:

```
#recuadro1{
  left:30px;
  position:relative;
  border-radius: 20px;
  background-color:#ddd;
  width:200px;
  height:100px;
  padding:4px;
  -moz-animation-name: animacion1;
  -moz-animation-duration: 4s;
  -webkit-animation-name: animacion1;
  -webkit-animation-duration: 4s;
  -o-animation-name: animacion1;
  -o-animation-duration: 4s;
}
```

```
@-moz-keyframes animacion1 {
  from {
    left:30px;
    background-color:#ddd;
  }
  to {
    left:300px;
    background-color:#ff0;
  }
}
```

```
@-webkit-keyframes animacion1 {
  from {
    left:30px;
    background-color:#ddd;
  }
  to {
    left:300px;
    background-color:#ff0;
  }
}
```

```
@-o-keyframes animacion1 {
  from {
    left:30px;
    background-color:#ddd;
  }
  to {
    left:300px;
```



```
background-color:#ff0;  
}  
}
```

Como podemos ver inicializamos las propiedades animation-name con el nombre de la animación:

```
-moz-animation-name: animacion1;  
-webkit-animation-name: animacion1;  
-o-animation-name: animacion1;
```

Y la propiedad animation-duration con el tiempo duración de la animación:

```
-moz-animation-duration: 4s;  
-webkit-animation-duration: 4s;  
-o-animation-duration: 4s;
```

La sintaxis para especificar la animación propiamente dicha es idéntica para cada navegador pero como debemos utilizar el prefijo luego creamos tres:

```
@-moz-keyframes animacion1 {  
  from {  
    left:30px;  
    background-color:#ddd;  
  }  
  to {  
    left:300px;  
    background-color:#ff0;  
  }  
}
```

En la parte del from indicamos el estado inicial del elemento (en este caso 30 píxeles respecto a la izquierda y de color gris (#ddd)), luego en la sección del to indicamos los valores finales que debe tomar el elemento (300 píxeles respecto a la izquierda y de color amarillo(#ff0))



# DOCUMENTACIÓN

## DOCUMENTACIÓN

- Características CSS:  
[https://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada)
- API CSS: <https://developer.mozilla.org/en-US/docs/Web/API/CSS>
- Guia oficial: <https://www.w3c.es/Divulgacion/GuiasReferencia/CSS21/>

## DOCUMENTACIÓN ADICIONAL

- Tutorial: <https://www.w3schools.com/css/>
- Libros gratuitos de CSS: <https://uniwebsidad.com/libros/css>

## HERRAMIENTAS UTILIZADAS EN EL CURSO

- Visual Studio Code: <https://code.visualstudio.com/>

[illegible]

[www.iconotc.com](http://www.iconotc.com)

