

# Efectos visuales

## Ejemplo-1: display

### Un ejemplo de em puesto como bloque

Supongamos que tenemos el siguiente código en la hoja de estilo:

```
em {display:block}
```

Se aplica a este texto:

```
<h2>Un título con una <em>palabra</em> enfatizada</h2>  
  
<p>Y un párrafo con otra <em>palabra</em> enfatizada</p>
```

Obteniendo el siguiente resultado:

**Un título con una  
*palabra*  
enfatizada**

Y un párrafo con otra  
*palabra*  
enfatizada

## Ejemplo-2: posicionamiento relativo

### Position: relative

(P1) Con esta propiedad desplazamos un elemento de la posición que ocuparía normalmente.

(P2) A continuación, vamos a repetir este párrafo y el anterior (marcados P1 y P2) pero usaremos las siguientes propiedades para P2:

```
.p2 {position: relative;  
top: -2em;  
left: 80px; }
```


(P1) Col (P2) A continuación, vamos a repetir este párrafo y el anterior (marcados P1 y P2) pero usaremos las siguientes propiedades para P2:

```
.p2 {position: relative;  
top: -2em;  
left: 80px; }
```

Podemos observar en el segundo caso cómo P2 ha sido desplazado de su posición normal. Con `top: -2em` y `left: 80px`, lo movemos 2em hacia arriba (superponiéndolo en parte con P1) y 80px hacia la derecha. Sin embargo, no afecta la posición del elemento siguiente (el elemento PRE, en rojo) que viene a continuación y que mantiene su ubicación normal.

# Ejemplo-3: posicionamiento absoluto

## Desplazamientos: top, right, bottom, left

 Para ejemplificar estas propiedades, vamos a ubicar esta imagen dentro de un elemento DIV. A su vez, ese DIV estará contenido por otro -con fondo gris- al que le hemos asignado la propiedad `position: relative`, de modo que resulte el *bloque de contención* para los de la imagen.

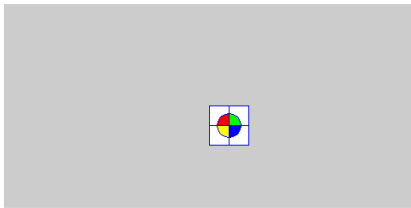
Para indicar esto, ponemos el siguiente código:

```
<div class="cont">
  <div id="ej1">
    
  </div>
</div>
```

Veamos ahora como se comportará el código anterior según los diferentes estilos que indiquemos. Así si ponemos:

```
ej1 {position: absolute;
     top: 50%;
     left: 50%;
}
```

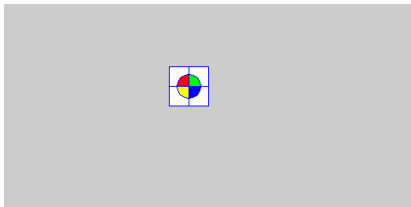
Obtendremos lo siguiente:



La imagen no aparece perfectamente centrada a pesar de haber usado un porcentaje de 50%. Esto es así porque el elemento que contiene la imagen no se alinea por su centro sino por su borde superior (para `top`) y por su borde izquierdo (para `left`).

Veamos el mismo porcentaje pero con `bottom` y `right`:

```
DIV {position: absolute;
     bottom: 50%;
     right: 50%;
}
```



En este segundo caso, es el vértice inferior derecho de la imagen el que queda ubicado exactamente en el centro del bloque de contención.

Con el uso de valores negativos podemos hacer que la imagen se ubique por fuera de su bloque de contención.

```
DIV {position: absolute;
     bottom: -23px;
     left: -23px;
}
```



Por eso, hay que tener cuidado al usar posicionamiento absoluto con valores que no sean porcentajes

# Ejemplo-4: capas

## Superponiendo capas con la propiedad 'z-index'

Esta es la capa numero 1.  
Esta es la capa numero 2.  
Puede que este por encima o por debajo de la capa numero 2.

En el ejemplo anterior, la capa número 1 no tiene valor asignado en z-index, pero la capa número 2 tiene un z-index de 1.

Lo hacemos al revés:

Esta es la capa numero 1.  
Esta es la capa numero 2.  
Puede que esté por encima o por debajo de la capa numero 2.

Ahora, la capa número 1 tiene valor de z-index de 2, mientras la número 2 sigue con un valor de 1.

Lo único que hemos hecho es añadir a cada elemento, dentro de la definición de la etiqueta STYLE un valor para la propiedad z-index. La capa con el valor mayor se superpondrá a las demás.

[Ver Código Fuente](#)

[Cerrar esta ventana](#)

# Ejemplo-5: float y clear

## La propiedad 'float'

El efecto de esta propiedad es bastante simple pero resulta muy útil para que el diseño de la página se adapte al monitor del usuario. Lo que hace la propiedad es quitar al elemento del flujo normal de la página y colocarlo a la izquierda o derecha del resto del contenido de su elemento padre (lo hace "flotar" hacia uno u otro costado).

En este ejemplo, el título tiene `float:none` y mantiene su posición normal.

**La propiedad 'float'** El efecto de esta propiedad es bastante simple pero resulta muy útil para que el diseño de la página se adapte al monitor del usuario. Lo que hace la propiedad es quitar al elemento del flujo normal de la página y colocarlo a la izquierda o derecha del resto del contenido de su elemento padre (lo hace "flotar" hacia uno u otro costado).  
En este ejemplo, el título tiene `float:left` y se coloca a la izquierda de este párrafo.

El efecto de esta propiedad es bastante simple pero resulta muy útil para que el diseño de la página se adapte al monitor del usuario. Lo que hace la **La propiedad 'float'** propiedad es quitar al elemento del flujo normal de la página y colocarlo a la izquierda o derecha del resto del contenido de su elemento padre (lo hace "flotar" hacia uno u otro costado).  
En este ejemplo, el título tiene `float:right` y se coloca a la derecha de este párrafo.


## La propiedad clear


Esta propiedad complementa a la anterior. Con `clear` se especifica que un elemento *no* admite un flotante sobre su izquierda, sobre su derecha o a ambos lados. También se la puede usar en un elemento flotante para impedir que otros flotantes puedan quedar sobre algunos de sus costados.

▶ Veamos este ejemplo: el div con la imagen de la flecha tiene la propiedad `float:left`, de modo que se ubicará a la izquierda de este párrafo.



Ahora, el div con la imagen de la flecha tiene la propiedad `float:left` pero este párrafo tiene `clear:left`, de modo que no admite un elemento flotante a su izquierda. El resultado es que la imagen se ubica por encima del párrafo.

▶  Ahora tenemos dos elementos div con dos imágenes, ambos tienen la propiedad `float:left`. El efecto es que ambas se ubican a la izquierda de este párrafo. Debido a las reglas para los flotantes, primero se ubica la flecha sobre la izquierda y luego la mancha (que también flota a la izquierda) se ubica sobre su derecha.

▶  Se repite el caso anterior, pero ahora el div con la imagen azul tiene la propiedad `clear:left`, de modo que no admite a otro flotante sobre su izquierda. El resultado es que ambas imágenes quedan sobre la izquierda de este párrafo, pero esta vez una imagen encima de la otra.

# Ejemplo-6: overflow

## 'overflow'

La propiedad 'overflow' se aplica a los elementos de bloque (por ejemplo: `blockquote`, `div`, `h1/h6`, `hr` (línea horizontal), `p`, `pre`) y también a los elementos reemplazados (como `img`). Primero vamos a crear un elemento `pre` que tiene asignadas las siguientes propiedades:

```
pre { background-color: #ffffcc;
  color: blue;
  border: 1px solid green;
  width: 50%;
  height: 12em; }
```

```
a:      Enlace de hipertexto o destino de un link
abbr:   Abreviatura
acronym: Acrónimo
address: Dirección
area:   Área de un mapa de imagen
b:      Texto en negrita
```

El elemento ocupa un 50% del ancho de la página y una altura aproximada al doble del texto que contiene. Repetimos el ejemplo, esta vez reduciendo su tamaño a la mitad (`width: 25%; height: 6em`):

```
a:      Enlace de hipertexto o destino de un link
abbr:   Abreviatura
acronym: Acrónimo
address: Dirección
area:   Área de un mapa de imagen
b:      Texto en negrita
```

Esto es lo que sucede: el fondo de color indica el tamaño que tiene el elemento; en el segundo caso (según la resolución de su monitor, puede suceder también en el primer caso) el contenido sobresale de ese fondo. En esos casos, se dice que el contenido está *desbordando* la caja que lo contiene.

Como el valor inicial para la propiedad 'overflow' es 'visible', el texto que desborda es procesado fuera de la caja del elemento `pre`. Ahora, repetimos el segundo ejemplo aplicando los otros valores que puede tener 'overflow':

Con `overflow: hidden`

```
a:      Enlace de hipertexto o
abbr:   Abreviatura
acronym: Acrónimo
address: Dirección
area:   Área de un mapa de ima
```

Con `overflow: scroll`

```
a:      Enlace de hipertexto
abbr:   Abreviatura
acronym: Acrónimo
address: Dirección
```

Con `overflow: auto`

```
a:      Enlace de hipertexto
abbr:   Abreviatura
acronym: Acrónimo
address: Dirección
```

El valor `hidden` provoca que el texto que desborda la caja sea recortado, de modo que sólo pueda verse una parte del mismo. Los valores `scroll` y `auto` también recortan el texto pero aparecen unas barras de desplazamiento que permiten hacer un scroll para ver las partes ocultas.

# Ejemplo-7: display con scripts

En este ejemplo debéis estudiar la interacción entre `display` y un sencillo script, que permite a través de un clic del ratón, que aparezca y desaparezca un párrafo. Dada la dificultad del ejercicio, os muestro en la página web, el código html, aunque debéis validarlo, partiendo de los siguientes errores:

**Result:** Failed validation, 40 errors

## Display: none

Vamos a usar dos imágenes para demostrar la propiedad display:

Imagen 1. `Display:inline`



Imagen 2. `Display:none`

La segunda imagen no es visible y el espacio es ocupado por el contenido que está a continuación (este párrafo y los siguientes).

### Utilización con scripts

Este es un ejemplo muy sencillo del uso que puede tener el valor *none* para la propiedad *display* en JavaScript. Se trata de un párrafo normal que al recibir un click esconde o muestra alternativamente el contenido de un elemento DIV.

HACER CLICK AQUÍ PARA OCULTAR O MOSTRAR

Este párrafo está contenido dentro de un DIV al cual, mediante JavaScript, se le cambia la propiedad display para que sea visible si está oculto o para que desaparezca si está visible. Cuando no es visible, el lugar que ocupaba es llenado por el contenido que está a continuación en la página.

Estas son las propiedades usadas:

```
<p style="cursor:s-resize" onClick="Alternar(seccion1)">
  HACER CLICK AQUÍ PARA OCULTAR O MOSTRAR
</p>
<div id="seccion1" style="display:">
  Este párrafo está contenido dentro de un div...
</div>
```

Cuando se señala el primer párrafo, el cursor del ratón se transforma en una doble flecha [`cursor: s-resize`] para indicar visualmente la acción. Al hacer un click en el párrafo se llama a la función `Alternar` del script y se identifica al div que se va a ocultar o mostrar [`onClick="Alternar(seccion1)"`].

El div tiene un nombre único [`ID="seccion1"`] que se usa como parámetro al llamar a la función `Alternar` y la propiedad `display` sin ningún valor.

Ahora, éste es el script:

```
function Alternar(Seccion) {
  if (Seccion.style.display=="none") {Seccion.style.display=""}
  else {Seccion.style.display="none"}
}
```

Lo que hace es verificar si el elemento que se pasa como argumento (en este ejemplo es `seccion1`) tiene el valor *none* para la propiedad `display` [`if (Seccion.style.display=="none")`] y en ese caso le elimina el valor *none* [`{Seccion.style.display=""}`], en caso contrario le pone el valor *none* al elemento [`else {Seccion.style.display="none"}`].

Este script sólo funciona en IE Explorer 4 ó superior y no pretende enseñar el uso de JavaScript sino una posible aplicación de la propiedad `display`. Lo interesante en este caso es que casi todo puede resolverse mediante el uso de estilos, por ejemplo el párrafo que actúa como disparador de la función tiene asignada la propiedad `cursor:s-resize` para que haya alguna indicación visual sin la necesidad de usar un link en el párrafo. De igual modo, el script se resuelve sencillamente cambiando el valor de la propiedad `display` sin necesidad de apelar a ningún otro recurso.

Cerrar

## Ejemplo-8: overflow y clip

Con la imagen del instituto, **ies.jpg**, utiliza "overflow y clip" junto con el posicionamiento absoluto para conseguir la siguiente composición, observa que en la parte inferior central de la imagen, aparece otra imagen con el nombre del centro, más reducida. Tienes que utilizar la imagen 2 veces, una de ellas recortada con overflow y clip.

# Utilizar el navegador Internet Explorer



## Ejemplo-9: Position absolute y relative

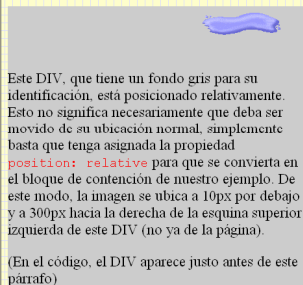
### Position: absolute

En este ejemplo vamos a usar un elemento DIV que contiene la imagen de una mancha azul con las siguientes propiedades:

```
DIV#ejemplo { position: absolute;
              top: 10px;
              left: 300px;}
```

Si observa el código de esta página, el DIV que contiene la imagen está ubicado justo antes de este párrafo, pero, como el bloque de contención de este DIV es la página (el elemento BODY), la imagen aparecerá a 10px por debajo y a 300px hacia la derecha de la esquina superior izquierda de la página.

Ahora bien, si el DIV de nuestro ejemplo se encontrara incluido dentro de otro DIV posicionado, este último se convertiría en el bloque de contención y entonces la imagen se ubicaría con relación a él (y no con respecto a la página).

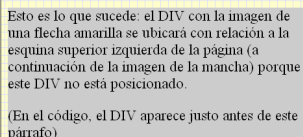


Este DIV, que tiene un fondo gris para su identificación, está posicionado relativamente. Esto no significa necesariamente que deba ser movido de su ubicación normal, simplemente basta que tenga asignada la propiedad `position: relative` para que se convierta en el bloque de contención de nuestro ejemplo. De este modo, la imagen se ubica a 10px por debajo y a 300px hacia la derecha de la esquina superior izquierda de este DIV (no ya de la página).

(En el código, el DIV aparece justo antes de este párrafo)

Para confirmar, vamos a utilizar un elemento DIV como el anterior pero sin definir su posición y dentro de él otro DIV con la imagen de una flecha y estas propiedades:

```
DIV#ejemplo2 { position: absolute;
               top: 10px;
               left: 430px;}
```



Esto es lo que sucede: el DIV con la imagen de una flecha amarilla se ubicará con relación a la esquina superior izquierda de la página (a continuación de la imagen de la mancha) porque este DIV no está posicionado.

(En el código, el DIV aparece justo antes de este párrafo)

Esto debe quedar en claro: la ubicación de un elemento con `position: absolute` se fija con respecto a la esquina superior izquierda de la página, a menos que ese elemento tenga un antepasado posicionado (es decir, con un valor `relative`, `absolute` o `fixed` para `position`), en cuyo caso su posición se fija con respecto a la esquina superior izquierda de ese antepasado.

Se debe tener en cuenta que el valor `absolute` (como así también `fixed`) provocan que el elemento no tenga ninguna influencia sobre los elementos siguientes. Veamos un ejemplo: a continuación tenemos cuatro párrafos, el primero y el tercero tienen la propiedad `position`. Para el primero, el valor de `position` es `relative` y para el tercero es `absolute`. No se usa ninguna otra propiedad (salvo el color, para diferenciarlos) de modo que ambos párrafos mantienen su posición dentro del documento.

Un párrafo con `position: relative`

Otro párrafo

Un párrafo con `position: absolute`

Lo que sucede es que el tercer párrafo (`position: absolute`), a pesar de mantener la misma posición, ha sido quitado del flujo normal de la página y el párrafo siguiente ocupa su lugar (provocando la superposición de ambos).

Cerrar

# Ejemplo-10: Position fixed

## Position: fixed

Vamos a crear un elemento DIV con las siguientes propiedades:

```
#cabecera { position: fixed;
            width: 100%;
            height: 48px;
            top: 0;
            left: 0;
            background-color: #99FFFF; }
```

Este es el DIV que se ubicará en la cabecera de la página.

En el código de la página, el DIV con `position:fixed` está ubicado junto antes de este párrafo. El resultado es un rectángulo celeste que abarca todo el ancho de la página ubicado en la parte superior de la misma. Al hacer un desplazamiento (scroll), el rectángulo se mantiene en su posición, lográndose el mismo efecto que con el uso de marcos (frames).

...

...

**Observa las diferencias al visualizar el documento con IE o con Firefox**

Este es el DIV que se ubicará en la cabecera de la página.

## Position: fixed

Vamos a crear un elemento DIV con las siguientes propiedades:

```
#cabecera { position: fixed;
            width: 100%;
            height: 48px;
            top: 0;
            left: 0;
            background-color: #99FFFF; }
```

En el código de la página, el DIV con `position:fixed` está ubicado junto antes de este párrafo. El resultado es un rectángulo celeste que abarca todo el ancho de la página ubicado en la parte superior de la misma. Al hacer un desplazamiento (scroll), el rectángulo se mantiene en su posición, lográndose el mismo efecto que con el uso de marcos (frames).

...

...

**Observa las diferencias al visualizar el documento con IE o con Firefox**

# Ejercicio-11

Incluir un fichero html donde se usen ejemplos de efectos visuales. Por tanto, debéis usar las etiquetas `<style>` `</style>` y no una hoja de estilos externa. No incluyáis gráficos, ya que eso requeriría otros ficheros. En el fichero debéis explicar qué estilos estáis utilizando para conseguir el efecto. Tenéis que consultar cada uno de los 30 efectos del apartado [Recursos visuales](#) del capítulo 5 y realizar el vuestro. También podéis utilizar efectos visuales que encontréis en Internet, siempre basados en CSS.