

# Tema 19. Propiedades de visualización.

## Indice

1. Preámbulo.....	2
2. Posicionamiento.....	2
2.1. Posicionamiento estático.....	2
2.2. Posicionamiento relativo.....	3
2.3. Posicionamiento absoluto.....	5
2.4. Posicionamiento fijo.....	7
3. Flotar.....	8
4. Despejar.....	10
5. Superposición.....	11
6. Desbordamiento.....	13
7. Recorte.....	15
8. Visualización.....	17
9. Visibilidad.....	19
10. Cursor del ratón.....	20
11. Numeración automática.....	23

## 1. Preámbulo

En el pasado, el posicionamiento exacto de un elemento en Html se convertía por lo general en una pesadilla y, generalmente, se realizaba a base de recurrir a un puzzle de tablas anidadas.

Las hojas de estilo CSS proporcionan ahora las herramientas adecuadas para posicionar, con una precisión de píxel, cualquier elemento en su documento Html. Liberadas en sus inicios bajo el término CSS-P, estas propiedades de posicionamiento se incluyen desde la especificación CSS 2.

Los elementos posicionados mediante las hojas de estilo CSS se podrán generar dinámicamente gracias a JavaScript.

## 2. Posicionamiento

Un elemento puede posicionarse de cuatro formas distintas.

### 2.1. Posicionamiento estático

Es el posicionamiento normal del elemento según la forma habitual de proceder del navegador.

El posicionamiento estático se determina mediante el atributo:

```
position: static;
```

El diseñador no tiene el control. El elemento no puede posicionarse ni reposicionarse y su visibilidad no puede modificarse. Del mismo modo, no es posible utilizar JavaScript para cambiar la posición del elemento.

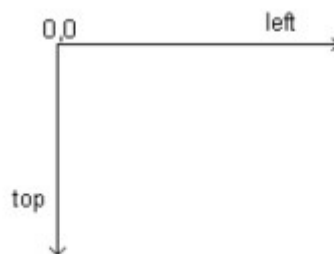
## 2.2. Posicionamiento relativo

Es el posicionamiento de un elemento respecto a su posición normal o estática.

Este elemento permanece en el flujo de los datos aunque está, en cierta medida, descentrado respecto a su posición normal.

La posición se define mediante las coordenadas (x,y) donde:

- ♦ **x** es la distancia respecto al borde izquierdo del elemento padre o de la ventana del navegador (eje horizontal). De este modo, `left` determina la distancia entre la parte izquierda del elemento y el borde izquierdo de la página y `right` la distancia entre la parte derecha del elemento y el borde derecho de la página.
- ♦ **y** es la distancia respecto al borde superior del elemento padre o de la ventana del navegador (eje vertical). De este modo, `top` determina la distancia entre el borde superior del elemento y el borde superior de la página y `bottom` la distancia entre el borde inferior del elemento y el borde inferior de la página.



El posicionamiento relativo se determina por:

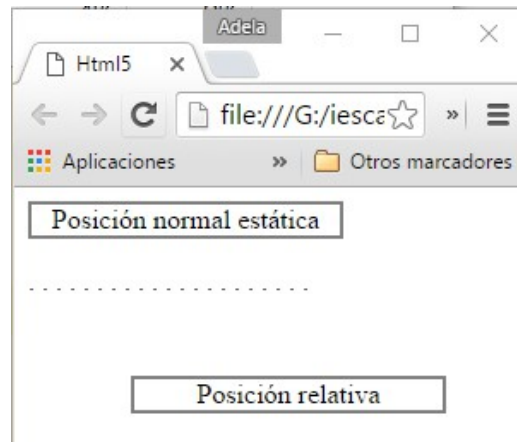
<code>position: relative;</code>	<code>left: valor o %;</code>
	<code>top: valor o %;</code>
	<code>right: valor o %;</code>
	<code>bottom: valor o %;</code>

## Comentarios

- ◆ En la práctica, basta con informar una sola especificación en el eje horizontal (left o right) y otra sobre el eje vertical (top o bottom).
- ◆ Los valores de top, left, right y bottom pueden ser negativos.

## Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.normal { width: 180px;
          border: 2px solid gray;
          text-align: center;}
.relativa { position: relative;
            top: 30px;
            left: 60px;
            width: 180px;
            border: 2px solid gray;
            text-align: center;}
</style>
</head>
<body>
<div class="normal">
Posición normal estática
</div>
<p>.....</p>
<div class="relativa">
Posición relativa
</div>
</body>
</html>
```



## 2.3. Posicionamiento absoluto

El posicionamiento absoluto crea un elemento independiente del resto del documento. Los elementos definidos con posición absoluta se retiran del flujo normal y se posicionan en el lugar exacto definido por el diseñador.

La posición se define mediante las coordenadas (x,y) donde:

- **x** es la distancia respecto al borde izquierdo del elemento padre o de la ventana del navegador (eje horizontal). De este modo, `left` determina la distancia entre el borde izquierdo del elemento y el borde izquierdo de la página y `right`, la distancia entre el borde derecho del elemento y el borde derecho de la página.
- **y** es la distancia respecto al borde superior del elemento padre o de la ventana del navegador (eje vertical). De este modo, `top` determina la distancia entre el borde superior del elemento y el borde superior de la página y `bottom` la distancia entre el borde inferior del elemento y el borde inferior de la página.

El posicionamiento absoluto se determina por:

```
position: absolute;
```

```
left: valor o %;
```

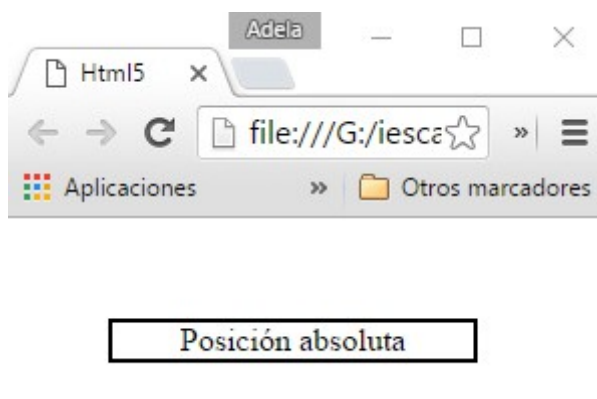
```
top: valor o %;
```

```
right: valor o %;
```

bottom: valor o %;

## Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.absoluta { position: absolute;
            left: 50px; top: 50px;
            width: 180px;
            border: 2px solid black;
            text-align: center;}
</style>
</head>
<body>
<div class="absoluta">
Posición absoluta
</div>
</body>
</html>
```



## Comentarios

- El posicionamiento absoluto tiene el riesgo de no mostrar correctamente la página en resoluciones de pantalla diferentes a la utilizada en tiempo de diseño puesto

que los elementos que se hayan posicionado mediante esta propiedad pueden superponerse sobre los elementos del flujo normal de la página.

- Conviene recordar que los navegadores agregan, por defecto, un margen al cuerpo de la página Html y que este margen por defecto varía de un navegador a otro. Por ello es prudente, en caso de usar posicionamiento absoluto, especificar los márgenes de la etiqueta <body>.

## 2.4. Posicionamiento fijo

El posicionamiento fijo también crea un elemento independiente del que se puede definir la posición exacta. En este caso el elemento se mantiene fijo aunque el documento se desplace o se deslice en pantalla.

<code>position: fixed;</code>	<code>left: valor o %;</code> <code>top: valor o %;</code> <code>right: valor o %;</code> <code>bottom: valor o %;</code>
-------------------------------	--

Los navegadores han ignorado durante mucho tiempo este tipo de posicionamiento fijo, aunque a día de hoy ya está perfectamente integrado en los navegadores de nuestro estudio.

### Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.fija { position: fixed;
        top: 50%;
        left: 50px;
        width: 180px;
```

```
border: 2px solid black;  
text-align: center;}  
</style>  
</head>  
<body>  
<div class="fija">  
Posición fija  
</div>  
</body>  
</html>
```



### 3. Flotar

La propiedad float permite retirar un elemento de tipo caja del flujo normal para situarlo lo más a la derecha o lo más a la izquierda posible dentro de su elemento padre, es decir su contenedor.

float:	right; o bien left; o bien none;
--------	--

#### Comentarios

- El valor `right` alinea a la derecha el elemento indicado, empujando a los demás



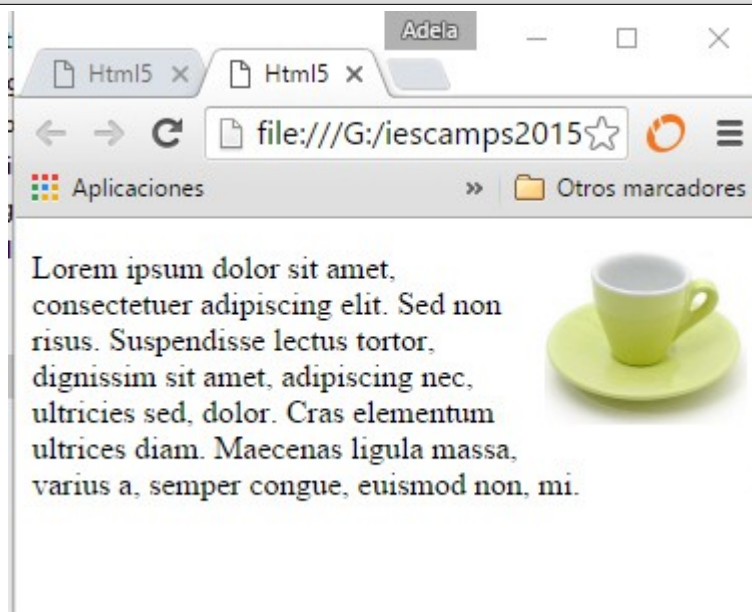
elementos hacia la izquierda.

- El valor `left` alinea a la izquierda el elemento indicado, empujando a los demás elementos hacia la derecha.
- El valor `none` no especifica nada y deja la gestión al navegador.
- La posición `float` no puede aplicarse en caso de posicionamiento absoluto.

## Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<div style="float:right;">

</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor. Cras elementum ultrices
diam. Maecenas ligula massa, varius a, semper congue, euismod non,
mi.</p>
</div>
</body>
</html>
```



El elemento contenedor es la división box. La propiedad de estilo `float: right;` fuerza a la imagen a posicionarse en la parte derecha del contenedor. El texto se distribuye a la izquierda.

## 4. Despejar

La propiedad `clear` permite anular el efecto introducido por la propiedad `float`.

<code>clear:</code>	<code>right;</code>  <code>o bien left;</code>  <code>o bien both;</code>  <code>o bien none;</code>
---------------------	--

### Comentarios

- ◆ El valor `right` anula los elementos flotantes a la derecha.
- ◆ El valor `left` anula los elementos flotantes a la izquierda.
- ◆ El valor `both` anula los elementos flotantes de ambos lados.
- ◆ El valor `none` anula los elementos flotantes.

### Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<div style="float:right;">

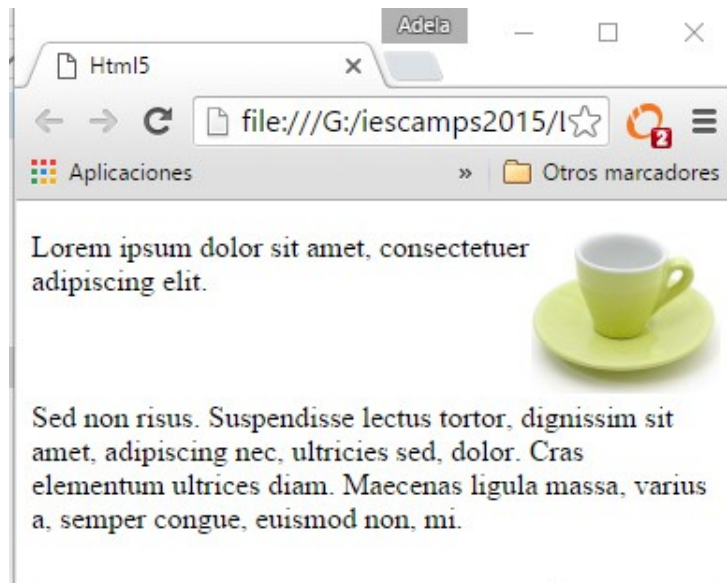
</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
<p style="clear:right">Sed non risus. Suspendisse lectus tortor,
dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras
```

*elementum ultrices diam. Maecenas ligula massa, varius a, semper  
congue, euismod non, mi.*

</div>

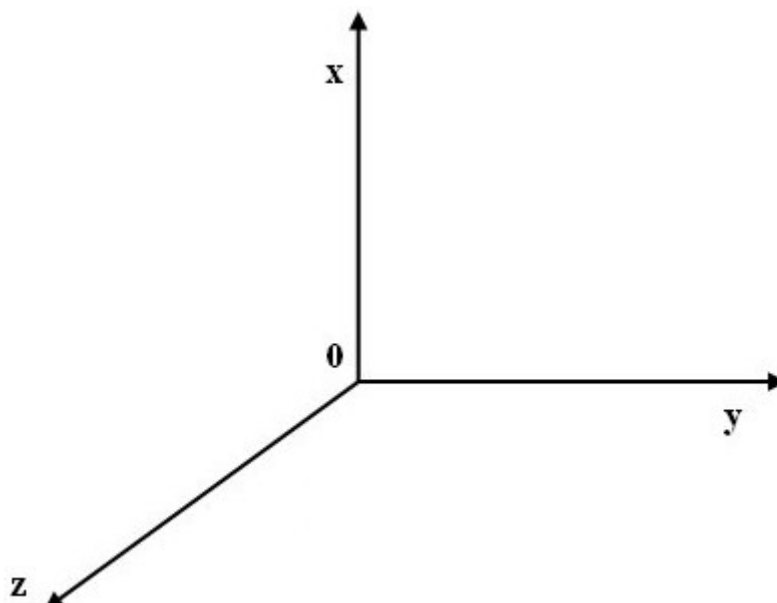
</body>

</html>



## 5. Superposición

La propiedad **z-index** agrega un eje de profundidad que permite posicionar los elementos los unos por encima o por debajo de los otros.



De este modo, si un elemento tiene la propiedad **z-index: 2** aparecerá por delante de otro elemento que tenga la propiedad **z-index: 0**.

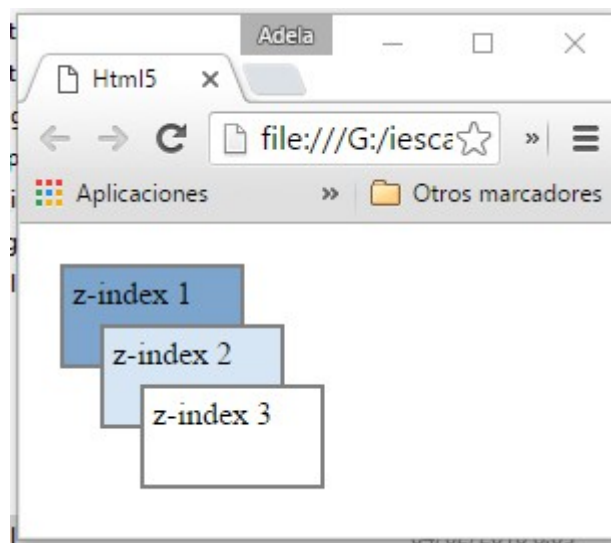
z-index:	un número entero (positivo);
----------	------------------------------

Esta propiedad sólo funciona con un posicionamiento absoluto de los elementos.

### Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.caja1 { position: absolute;
        left: 20px; top: 20px;
        width: 80px; height: 40px;
        padding: 4px;
        border: 2px solid gray;
        background-color: rgb(125,165,205);
        z-index: 1;}
.caja2 { position: absolute;
        left: 40px; top: 50px;
        width: 80px; height: 40px;
        padding: 4px;
        border: 2px solid gray;
        background-color: rgb(215,230,245);
        z-index: 2;}
.caja3 { position: absolute;
        left: 60px; top: 80px;
        width: 80px; height: 40px;
        padding: 4px;
        border: 2px solid gray;
        background-color: white;
        z-index: 3;}
</style>
</head>
<body>
<div class="caja1">
z-index 1
</div>
<div class="caja2">
```

```
z-index 2
</div>
<div class="caja3">
z-index 3
</div>
</body>
</html>
```



## 6. Desbordamiento

La propiedad **overflow** determina lo que debe hacer el navegador cuando un elemento es más grande que el elemento padre que lo contiene.

overflow:	hidden;
	o bien scroll;
	o bien visible;
	o bien auto;

### Comentarios

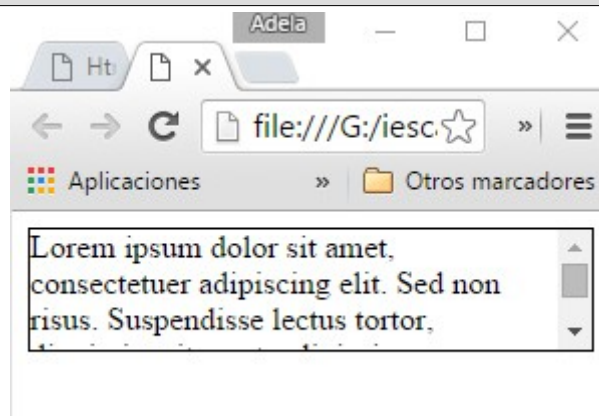
- ◆ Con el valor **hidden**, la parte que desborda se oculta, sin posibilidad de verla ni

acceder a ella.

- ◆ Con el valor **scroll**, la parte que desborda se oculta, aunque queda accesible mediante barras de desplazamiento.
- ◆ Con el valor **visible**, la parte que desborda se muestra, ignorando las especificaciones del elemento padre que lo contiene.
- ◆ Con el valor **auto**, se deja la gestión al navegador.

## Ejemplo

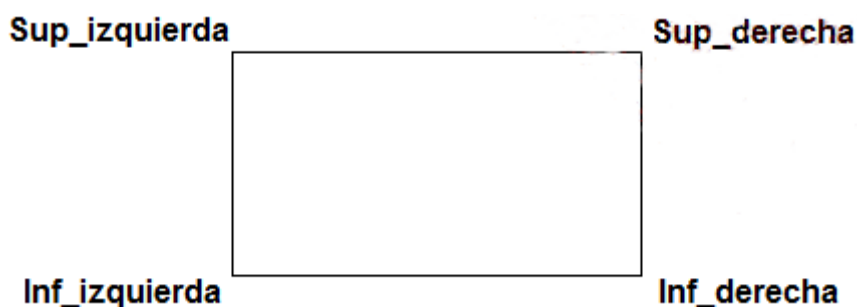
```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.caja { width: 280px; height: 60px;
        border: 1px solid black;
        overflow: auto;}
</style>
</head>
<body>
<div class="caja">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non
risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing
nec, ultricies sed, dolor. Cras elementum ultrices diam. </div>
</body>
</html>
```



## 7. Recorte

Esta propiedad clip determina la parte visible del elemento, generalmente una imagen. De este modo la imagen entera se incluirá en el documento aunque solamente una parte (rectangular) de la misma estará visible.

clip:	<pre>rect(sup_izquierda sup_derecha inf_derecha inf_izquierda);  auto;</pre>
-------	--



Por ejemplo:

```
clip: rect(10px 270px 190p 50px);
```

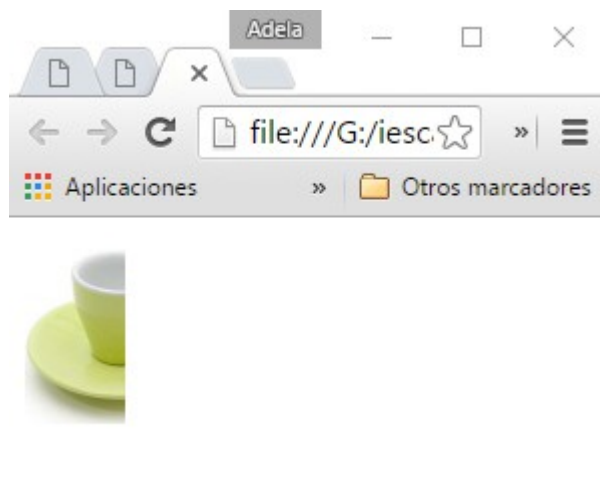
### Comentarios

- ◆ Las coordenadas del rectángulo recortado vienen dadas por las esquinas superior izquierda, superior derecha, inferior derecha e inferior izquierda respecto de la imagen inicial.
- ◆ Se recomienda incluir el elemento que se quiere recortar dentro de una etiqueta <div> o <span>.
- ◆ Esta propiedad solamente funciona con un posicionamiento absoluto del elemento.
- ◆ De momento sólo está implementada la forma rectangular.

## Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.recorte { position: absolute;
            clip: rect(0px 50px 115px 0px);}
</style>
</head>
<body>
<p></p>
<div class="recorte">

</div>
</body>
</html>
```





## 8. Visualización

La propiedad `display` permite controlar la visualización de los elementos en la página.

<code>display:</code>	<code>block;</code>  <code>o bien inline;</code>  <code>o bien none;</code>
-----------------------	---

En el capítulo Propiedades de caja - Elemento en bloque o en línea, ya hemos tratado la propiedad `display` que permite redefinir un elemento en línea como un elemento de tipo bloque y a la inversa (`display: block` y `display: inline`).

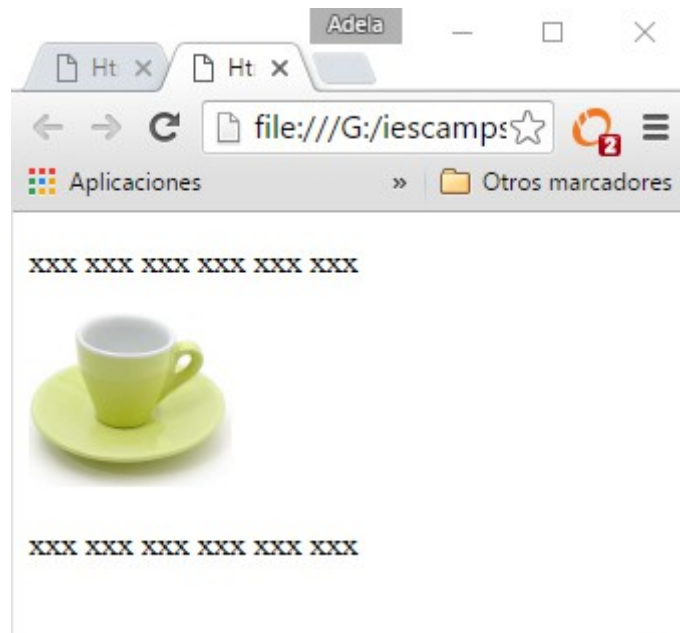
Es la propiedad `display: none` la que nos interesa especialmente aquí. Define un elemento que no se va a mostrar. Este elemento se retira del documento y de la visualización en la página. Véase la sección Visibilidad, a continuación, para comprender la diferencia respecto a la propiedad `visibility: hidden;`.

### Ejemplo 1

Visualización normal de la imagen.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<p>xxx xxx xxx xxx xxx xxx</p>
<div>

</div>
<p>xxx xxx xxx xxx xxx xxx</p>
</body>
</html>
```

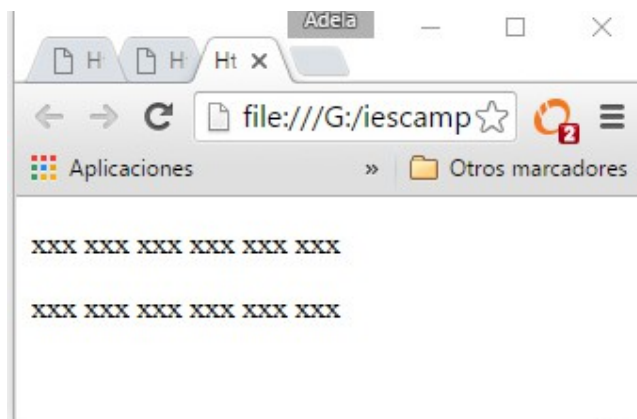


## Ejemplo 2

Con la propiedad display:none.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.displaynone { display: none;}
</style>
</head>
<body>
<p>xxx xxx xxx xxx xxx xxx</p>
<div class="displaynone">

</div>
<p>xxx xxx xxx xxx xxx xxx</p>
</body>
</html>
```



La captura de pantalla ilustra cómo la imagen se ha retirado, igual que si estuviera ausente en el código

## 9. Visibilidad

La propiedad CSS visibility determina si un elemento está visible u oculto.

visibility:	visible; o bien hidden;
-------------	----------------------------

### Comentarios

- ◆ El valor `visible` muestra el elemento.
- ◆ El valor `hidden` oculta el elemento.

Preste atención, pues a pesar de establecer el valor `hidden` para esta propiedad, el elemento ocupa todavía su hueco en el documento y, en consecuencia, se muestra un rectángulo en blanco donde estuviera situado el elemento. De este modo, se conserva el formato de página a pesar de la ausencia (aparente) de la imagen, y los demás elementos de la página conservan su posición relativa inicial. El valor `hidden` difiere así de la propiedad `display: none;` que elimina por completo el elemento de la página y no reserva sitio alguno.

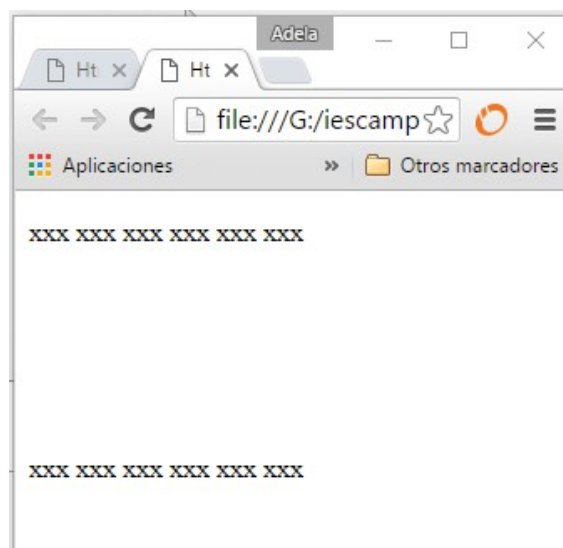
### Ejemplo

Con la propiedad `visibility: hidden`.

```
<!DOCTYPE html>
<html lang="es">
```

```
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.hidden { visibility: hidden;}
</style>
</head>
<body>
<p>xxx xxx xxx xxx xxx xxx</p>
<div class="hidden">

</div>
<p>xxx xxx xxx xxx xxx xxx</p>
</body>
</html>
```



La captura de pantalla muestra claramente cómo, aunque la imagen no está visible, sí se ha reservado el espacio asignado a la misma en el interior de la página.

## 10. Cursor del ratón















La propiedad cursor permite modificar el cursor del ratón.

cursor:	palabra clave;
---------	----------------

Las distintas palabras clave son:

- `pointer`: el cursor presenta un dedo que indica un enlace.
- `move`: el cursor indica un objeto que se puede desplazar.
- `e-resize`: cursor apuntando hacia el este.
- `ne-resize` cursor apuntando hacia el noreste.
- `nw-resize`: cursor apuntando hacia el noroeste.
- `n-resize`: cursor apuntando hacia el norte.
- `se-resize`: cursor apuntando hacia el sureste.
- `sw-resize`: cursor apuntando hacia el suroeste.
- `s-resize`: cursor apuntando hacia el sur.
- `w-resize`: cursor apuntando hacia el oeste.
- `text`: el cursor indica que es posible seleccionar el texto.
- `wait`: el cursor indica una progresión. A menudo se muestra un reloj de arena.
- `help`: el cursor indica una ayuda. A menudo se muestra un signo de interrogación.
- `default`: el cursor por defecto del sistema operativo, a menudo una flecha.
- `crosshair`: el cursor muestra una cruz.
- `progress`: el cursor muestra una flecha con un reloj de arena.
- `not-allowed`: el cursor muestra una señal de prohibido.
- `no-drop`: el cursor tiene forma de mano con un dedo desplegado y una señal de prohibido.
- `col-resize`: cursor compuesto por dos trazos verticales con una flecha a cada lado.
- `row-resize`: cursor compuesto por trazos horizontales.
- `auto`: el navegador determina por sí mismo el cursor según el contexto.
- `url`: especifica un archivo donde se encuentra la imagen que se desea usar como cursor. El archivo de imagen especificado en la URL debe tener el formato `cur` (cursor) o `ani` (cursor animado).

Algunos cursores tienen forma de imagen:

pointer	
move	
e-resize	
ne-resize	
nw-resize	
n-resize	
se-resize	
sw-resize	
s-resize	
w-resize	
text	
wait	
help	
crosshair	

La apariencia del cursor puede variar según el navegador y el sistema operativo.

### Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
h2 { cursor: help;}
</style>
```

```
</head>
<body>
<h2>Hojas de estilo CSS</h2>
</body>
</html>
```

## 11. Numeración automática

La especificación CSS2 ha introducido la noción de contadores (`counter`) en las propiedades de estilo. Gracias a ellos es posible gestionar, por ejemplo, una numeración automática de secciones o subsecciones.

<code>counter-reset:</code>	nombre del contador;
<code>counter-increment:</code>	valor de incremento;

### Comentarios

- ♦ La propiedad `counter-reset` crea o inicializa el contador indicado.
- ♦ La propiedad `counter-increment` incrementa el contador indicado.
- ♦ El paso de incremento es 1 por defecto. No obstante es posible especificar otro valor.

### Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
body { counter-reset: section;
      font-size: 50%;}
h1 { counter-reset: item;}
h1:before { counter-increment: section;
            content: "Capítulo " counter(section) ": ";}
h2:before { counter-increment: item;
```

```
        content: counter(section) "." counter(item) " ";}
</style>
</head>
<body>
<h1>Html</h1>
<h2>Html 4.0</h2>
<h2>Xhtml 1.0</h2>
<h2>Xhtml5</h2>
<h1>Scripts</h1>
<h2>JavaScript</h2>
<h2>Dhtml</h2>
<h1>Frameworks</h1>
<h2>jQuery</h2>
<h2>Dojo</h2>
</body>
</html>
```

Estas propiedades de estilo requieren algunas explicaciones.

```
body { counter-reset: section;}
```

Se inicializa el contador llamado section donde el campo de acción es el cuerpo del documento.

```
h1 { counter-reset: item;}
```

Se inicializa otro contador item donde el campo de acción son los títulos de nivel 1.

```
h1:before { counter-increment: section;
            content: "Capítulo " counter(section) ": " ;}
```

Por cada título **<h1>**, el contador section se incrementa en una unidad (counter-increment: section). La propiedad de estilo escribe delante de los títulos **<h1>** (h1:before) el contenido (content) indicado entre comillas. En este caso el texto "Capítulo" seguido del valor del contador section.

```
h2:before { counter-increment: item;
            content: counter(section) "." counter(item) " " ;}
```

Por cada título **<h2>**, el contador item se incrementa en una unidad (counter-increment: item). La propiedad de estilo escribe delante de los títulos **<h2>** (h2:before) el contenido (content) indicado entre comillas. En este caso el número del capítulo seguido



de un punto y seguido del valor del contador item.

