



MINISTERIO
DE EDUCACIÓN
Y CIENCIA

SECRETARÍA GENERAL
DE EDUCACIÓN
Y FORMACIÓN PROFESIONAL

DIRECCIÓN GENERAL
DE EDUCACIÓN,
FORMACIÓN PROFESIONAL
E INNOVACIÓN EDUCATIVA

CENTRO NACIONAL
DE INFORMACIÓN Y
COMUNICACIÓN EDUCATIVA

Edición HTML

Estilos CSS Diseño visual



SERVICIO DE
FORMACIÓN DEL
PROFESORADO

C/ TORRELAGUNA, 58
28027 - MADRID

Índice de contenido

Diseño visual.....	3
Cómo se muestran los elementos.....	3
Posición flotante.....	4
Zonas de flotación prohibidas.....	5
Posición de un elemento.....	8
Visibilidad.....	10
Recortes.....	11
Desbordamiento.....	12
Alineamiento vertical	13
Apariencia del cursor.....	14
Apilar capas.....	14
Absolutamente relativo.....	15

DISEÑO VISUAL

Bajo el epígrafe de diseño visual se incluyen propiedades referidas a la forma en que podemos controlar cómo se muestra un elemento, cómo aparecerá una imagen en relación a otro elemento, el posicionamiento de un elemento respecto a su posición habitual o con una posición fija dentro de la página, la determinación de la visibilidad de un elemento y el orden en el que se mostrarán las capas si estas se superponen, la creación de recortes para mostrar la información y cómo mostrarla cuando supera el espacio asignado.

CÓMO SE MUESTRAN LOS ELEMENTOS

Ya hemos visto que hay elementos en bloque y en línea y que su forma de construir el cuadro contenedor difiere bastante. Mediante el uso de CSS podemos indicarle al navegador que debe mostrar un elemento creando un tipo de cuadro distinto al que le correspondería por su naturaleza. En el editor de CSS de N|VU podrás acceder a esta propiedad dentro de la pestaña caja.

Propiedad: `display`

Valores: aunque podríamos considerar otros recogidos en la especificación vamos a tener en cuenta sólo `inline` y `block` para intercambiar la forma de presentación de las dos principales disposiciones.

Ejemplo 1

Si utilizo el selector `span` para marcar una palabra de una frase y aplicarle un estilo lo normal es que ese estilo genere un cuadro de elemento en línea ya que esa es la presentación predeterminada para los fragmentos de texto. A continuación se repite este mismo texto aplicándole la propiedad `display:block` a la palabra presentación.

Resultado

Si utilizo el selector `span` para marcar una palabra de una frase y aplicarle un estilo lo normal es que ese estilo genere un cuadro de elemento en línea ya que esa es la presentación predeterminada para los fragmentos de texto. A continuación se repite este mismo texto aplicándole la propiedad `display:block` a la palabra presentación.

Ejemplo 2

En el segundo ejemplo marcaremos los párrafos con `display:inline`

Este es el segundo párrafo. Ahora repetimos los dos párrafos correspondientes a este ejemplo y aparecerán uno a continuación de otro como si no hubiese salto de párrafo.

Resultado

En el segundo ejemplo marcaremos el segundo párrafo con `display:inline` Este es el segundo párrafo. Ahora repetimos los dos párrafos correspondientes a este ejemplo y aparecerán uno a continuación de otro como si no hubiese salto de párrafo.

POSICIÓN FLOTANTE

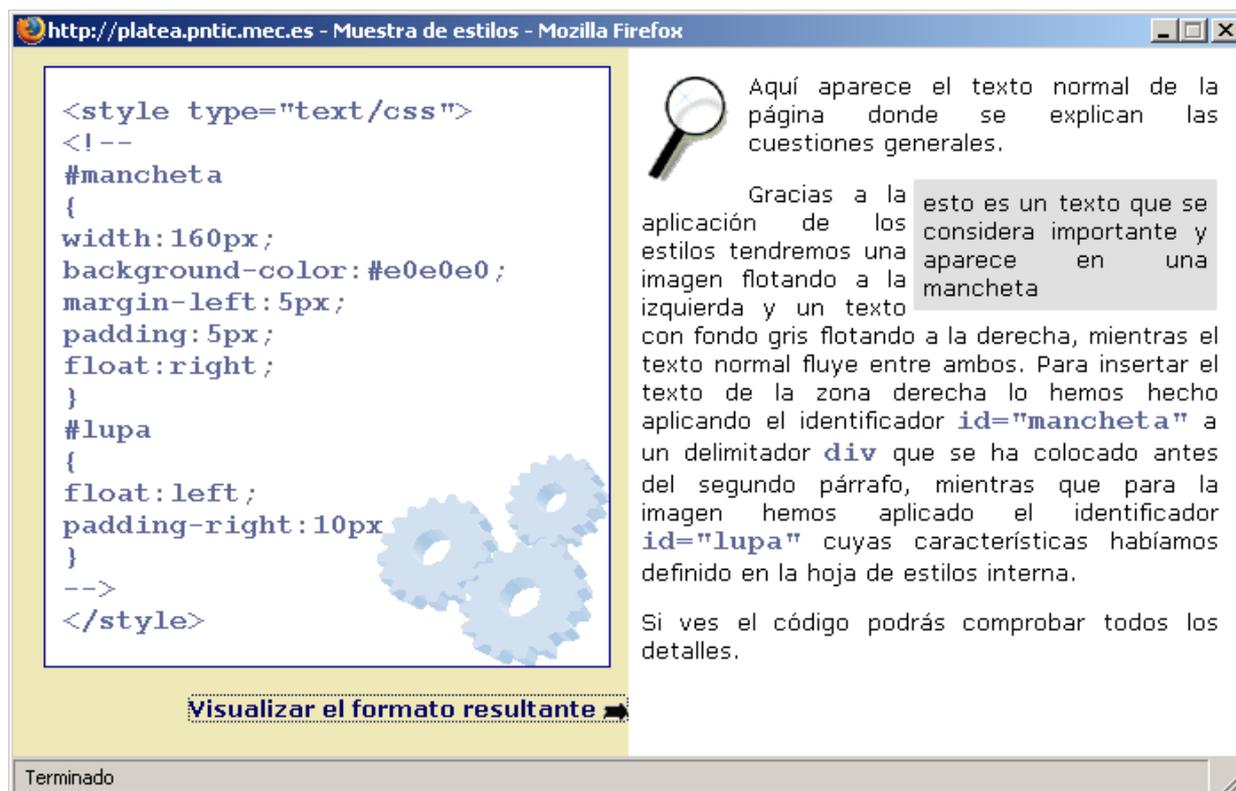
Se trata de una propiedad que encuentra su principal aplicación en las imágenes, aunque

también nos permitirá situar elementos textuales y conseguir interesantes efectos de inclusión, especialmente si utilizamos como complemento los fondos. En el editor de CSS de N|VU podrás acceder a esta propiedad dentro de la pestaña caja.

Propiedad: `float`

Valores: `left` hará flotar el elemento afectado en la zona izquierda, `right` en la derecha y `none` impedirá la flotación haciendo que el elemento se muestre en la posición que le corresponde en el flujo de la página.

Ejemplos: En el ejemplo que se muestra se han creado dos identificadores a los que se les ha aplicado la propiedad `float` y algunas más de fondos, márgenes y rellenos para que su presentación visual tenga "aire" entre las diferentes zonas.



http://platea.pntic.mec.es - Muestra de estilos - Mozilla Firefox

```
<style type="text/css">
<!--
#mancheta
{
width:160px;
background-color:#e0e0e0;
margin-left:5px;
padding:5px;
float:right;
}
#lupa
{
float:left;
padding-right:10px
}
-->
</style>
```

Aquí aparece el texto normal de la página donde se explican las cuestiones generales.

Gracias a la aplicación de los estilos tendremos una imagen flotando a la izquierda y un texto con fondo gris flotando a la derecha, mientras el texto normal fluye entre ambos. Para insertar el texto de la zona derecha lo hemos hecho aplicando el identificador `id="mancheta"` a un delimitador `div` que se ha colocado antes del segundo párrafo, mientras que para la imagen hemos aplicado el identificador `id="lupa"` cuyas características habíamos definido en la hoja de estilos interna.

Si ves el código podrás comprobar todos los detalles.

esto es un texto que se considera importante y aparece en una mancheta

Visualizar el formato resultante

Terminado

ZONAS DE FLOTACIÓN PROHIBIDAS

Si no nos interesa que a un determinado elemento se le añadan zonas flotantes podemos indicarlo mediante una propiedad específica. En el editor de CSS de N|VU podrás acceder a esta propiedad dentro de la pestaña caja.

Propiedad: `clear`

Valores: `left` impedirá la flotación de un elemento a la izquierda del que lleva la propiedad, `right` hará lo propio a la derecha y `both` afectará a ambos lados.

Ejemplos: Se repite el ejemplo anterior pero se ha evitado la flotación de la zona a la que habíamos llamado mancheta.

http://platea.pntic.mec.es - Muestra de estilos - Mozilla Firefox

```

<style type="text/css">
<!--
#mancheta
{
width:160px;
background-color:#e0e0e0;
margin-left:5px;
padding:5px;
float:right;
}
#lupa
{
float:left;
padding-right:10px
}
-->
</style>

```

Aquí aparece el texto normal de la página donde se explican las cuestiones generales.

esto es un texto que se considera importante y aparece en una mancheta

El código de este ejemplo era igual que el del anterior. Para insertar el texto de la zona derecha lo hemos hecho aplicando el identificador `id="mancheta"` a un delimitador `div` que se ha colocado antes del segundo párrafo, sin embargo, como hemos aplicado al segundo párrafo el estilo en línea `clear="right"` se ha impedido la flotación y el párrafo se ha colocado en el flujo normal tras el contenido del delimitador `div`.

Para la imagen hemos aplicado el identificador `id="lupa"` cuyas características habíamos definido en la hoja de estilos interna y flota porque no existe ninguna especificación de estilo que se lo impida.

Visualizar el formato resultante ➔

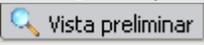
Si ves el código podrás comprobar todos los detalles.

Terminado

Antes de continuar recordemos cual es el proceso habitual de generación de páginas en una ventana de ordenador. Es lo que se denomina **flujo normal**:

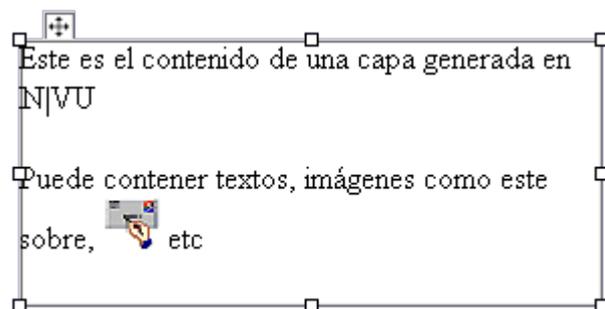
- Los elementos de bloque generan cajas de bloque que se sitúan verticalmente, una a continuación de otra en su orden de aparición en la página, comenzando por la parte superior de la ventana. El espaciado vertical entre un elemento y el siguiente se calcula tomando el mayor valor de margen superior o inferior de los elementos implicados.
- Los elementos en línea generan cajas de línea que se disponen horizontalmente uno a continuación de otro en su orden de aparición en la página, desde el inicio del bloque que las contiene. Si hay dos cajas de línea contiguas sus espaciados horizontales se mantienen, sumándose sus anchuras. Cuando una caja de línea ocupa más del espacio disponible se rompe automáticamente generando una nueva caja en el renglón inferior. En ese caso, los bordes, márgenes y rellenos no tienen aplicación en el borde de ruptura ni en el de inicio de la siguiente caja.

92 Vamos a comprobar con un ejercicio práctico lo que acabamos de comentar.

- Crea una página nueva con dos párrafos y aplícales a cada uno de ellos un estilo en línea que recoja las siguientes propiedades:
 - `margin:50px` y `border:thin solid blue` para el primero
 - `margin:10px` y `border:thin dotted red` para el segundo
 - también sería conveniente que al selector `body` le añadieras el estilo `margin:0px`
- Añade un nuevo párrafo sin características especiales.
- Modifica los valores de los márgenes en la pestaña  y alterna con  para comparar los efectos visuales obtenidos.
- Selecciona una palabra del primer párrafo encerrándola entre el delimitador `` y su cierre ``. Aplícale al delimitador el estilo `border:thin solid red`.
- Haz exactamente lo mismo con la palabra anterior o posterior a la que acabas de trabajar.
- Como tienes a la vista el resultado selecciona un grupo de palabras que incluyan un salto de línea y enciérralas entre `` y `` para aplicarles el mismo estilo de antes y ve a la pestaña  para comprobar la ausencia de los bordes al final y al principio de línea.
- Si quieres puedes guardar la página como **actividad92.html**

Para poder modificar la apariencia visual disponemos en CSS2 de la posibilidad de alterar el flujo normal especificando la posición de los elementos. Se trata de una propiedad muy potente y, aunque podemos aplicarla directamente sobre cualquier elemento, la recomendación sería que lo hicieras creando contenedores de bloque con la etiqueta `<div>` y especificando en ellos la posición.

Así pues, antes de ver los valores y los efectos de los mismos comentemos que N|VU dispone de un icono que nos permite generar una capa, que no es más que un contenedor en bloque, con una simple pulsación sobre el icono  que creará una capa vacía que se irá rellenando con el contenido que vayamos tecleando.



Si lo que pretendemos es crear una capa a partir de contenidos que ya estaban incluidos en la página basta con que los señalemos arrastrando y pulsemos el mismo icono para que se genere el contenedor para englobarlos a todos. Tanto en uno como en otro caso la capa generada lo es con posición absoluta, por lo que si lo único que pretendíamos era insertar una etiqueta `div` puede resultar más cómodo hacerlo utilizando el selector desplegable de tipo de párrafo o tecleando las etiquetas de apertura y cierre tras haber marcado el texto que queremos convertir en `div` y elegido **INSERTAR** .

Cuando nos situemos sobre una capa posicionada absolutamente el programa lo indicará haciendo aparecer sus límites y modificando el icono de las capas que aparecerá como .

Podremos moverla de un sitio a otro si pulsamos sobre  y arrastramos hasta la nueva posición o bien modificar su tamaño utilizando los tiradores de los vértices y el centro de los lados.

Para modificar los estilos que afectan a la capa bastará con hacer doble clic sobre su interior y modificar o añadir las propiedades que necesitemos.

- 93**
- Crea una página nueva con dos párrafos y selecciónalos
 - Pulsa  para crear una capa que los incluya.
 - Desplaza la capa hacia otra posición y haz clic para situar el cursor fuera de la capa.
 - Pulsa nuevamente  para crear una capa vacía y teclea un párrafo para darle contenido.
 - Haz un doble clic en el interior de una de las capas y utiliza la propiedad `background-color` para adjudicarle un color de fondo.
 - Arrastra la nueva capa de forma que se superponga sobre la anterior.
 - Utiliza los iconos  para modificar el orden en que se apilan. Puedes realizar todo el trabajo sobre la pestaña  Vista preliminar, pero para ver el efecto deberás pulsar sobre una zona fuera de las capas, de forma que no esté señalada ninguna de ellas.

POSICIÓN DE UN ELEMENTO

Habrás comprobado que al crear las capas N|VU ha introducido unos códigos de estilo. Como las hemos movido hasta situarlas en un punto concreto de la página ha utilizado el valor `absolute` que es uno de los posibles. En el editor de CSS de N|VU podrás acceder a esta propiedad dentro de la pestaña caja. El tema del posicionamiento es uno de los de mayor repercusión para el diseño de páginas con CSS. Puedes consultar una **ampliación de conceptos** que te ayudará a comprender este tema en profundidad. Tienes una ampliación de conceptos

Propiedad: `position`

Valores: Esta propiedad indicará la forma de calcular la posición del elemento utilizando los valores `absolute`, `relative`, `fixed` y `static`, que es el valor por defecto, pero para completar el cálculo será necesario añadir luego las propiedades `left`, `top`, `bottom` o `right` indicando las distancias en cualquiera de las unidades admitidas respecto a la caja contenedora. Estas distancias habra que indicarlas en el editor de CSS de N|VU dentro de la pestaña caja en la sección desplazamientos.

Ejemplos:

`Absolute`. **Ejemplo:**

http://platea.pntic.mec.es - Muestra de estilos - Mozilla Firefox

```

<style type="text/css">
<!--
#mancheta
{
width:160px;
background-color:#e0e0e0;
margin-left:5px;
padding:5px;
position:absolute;
bottom:40px;
right:100px;
}
-->
</style>

```

Si maximizas esta ventana comprobarás que la mancheta sigue colocándose a la misma distancia de los bordes inferior y derecho

este párrafo está situado de forma absoluta en la página, utilizando las coordenadas **bottom** y **right**

Visualizar el formato resultante

Terminado

Relative. La posición relativa se calcula a partir de la posición que ocuparía un elemento si siguiera el flujo normal. El resto de los elementos no se ajustan a la posición modificada por lo que puede ocurrir que el elemento se superponga sobre otros. **Ejemplo:**

http://platea.pntic.mec.es - Muestra de estilos - Mozilla Firefox

```

<p style = "position:relative ;
left:40px; top:-19px;
border:thin dotted orange">

```

Este sería el primer párrafo.

A continuación iría el segundo...

... y el tercero. Pero... repitamos la jugada

Este sería el primer párrafo.

A continuación iría el segundo...(pero ahora posición relativa). El desplazamiento debido a posición relativa ha provocado el desbordamiento del bloque por la derecha, ya que la caja del bloque del párrafo sigue teniendo la misma anchura, por haberse desplazado hacia su derecha necesita 40 píxeles de espacio suplementario.

... y el tercero.

Visualizar el formato resultante

Terminado

Fixed. Se trata de una variante de la posición absoluta cuyo efecto es similar al de las imágenes de fondo que se mantienen en una posición fija independientemente del desplazamiento de la página. Cuando se aplican a un medio impreso, formando algo similar a una cabecera, se repiten en todas las páginas. **Ejemplo:**

Este sería el primer párrafo que se mantendría fijo sin desplazarse aunque hubiera que desplazar el resto del contenido de la página

```

<style type="text/css">
<!--
.cabecera
{
position:fixed;
top:0px;
left:0px;
height:40px;
width:100%;
background-color:#e0e0e0;
}
-->
</style>

```

La propiedad **fixed** está claramente definida en la especificación CSS2, sin embargo su implementación en los navegadores es bastante irregular.

Si estás viendo esta página con Internet Explorer las situaciones pueden ser de lo más diversas, a pesar del valor **fixed** puede que el recuadro superior se desplace con el resto de la página, que no se ajuste, etc.

Sin embargo, cuando la página se visualiza con navegadores más respetuosos con las recomendaciones del W3C, como Mozilla, Konqueror o Galeón por ejemplo, podemos ver la zona de texto superior anclada al inicio de la ventana mientras el

[Visualizar el formato resultante](#)

javascript:void(0)

Static es el valor por defecto y no produce ninguna modificación en el posicionamiento del elemento, por lo que su misión puede ser más bien la de restablecer el flujo de la página después de haber situado algún elemento posicionado.



Internet Explorer 6 no soporta la posición **fixed**. Para poder utilizarla y que se vea en este navegador hay que aplicar un parche que veremos en la sección de recursos.

VISIBILIDAD

Como el nombre de la propiedad sugiere se refiere a la posibilidad de hacer que un elemento resulte visible o invisible. Su utilidad reside en la combinación con scripts que permitan mostrar u ocultar capas al realizar determinadas acciones con el ratón, pero este tema excede los contenidos de este curso, aunque puedes realizar alguna búsqueda sobre "javascript para mostrar y ocultar capas" para encontrar referencias. En el editor de CSS de N|VU podrás acceder a esta propiedad dentro de la pestaña caja.

Propiedad: `visibility`

Valores: Visible que es el valor por defecto o `hidden` para ocultarla

Ejemplos: Ver el código del ejemplo

- [Ver un ejemplo en funcionamiento en la documentación en HTML](#)
- El código para hacerlo funcionar es el que se muestra a continuación.

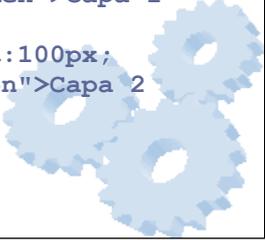
```

<html>
<head>
<title>Mostrar y ocultar capas </title>
<script language="Javascript">
function mostrar(nombreCapa) {
document.getElementById(nombreCapa).style.visibility="visible";
}
function ocultar(nombreCapa) {
document.getElementById(nombreCapa).style.visibility="hidden";
}
</script>
</head>

<body>
<ul>
<li>
<a href="#" onClick="mostrar('capa1')"> Mostrar la capa haciendo clic </a>
</li>
<li>
<a href="#" onClick="ocultar('capa1')"> Ocultar la capa 1 haciendo clic </a>

</li>
<li>
<a href="#" onMouseOver="mostrar('capa2')" onMouseOut="ocultar('capa2')">
Mostrar y ocultar la capa 2 al pasar sobre el enlace</a>
</li>
</ul>
<div id="capa1" style="position:absolute; width:100px; height:100px;
top:100px; left:100px; background-color:blue; visibility:hidden">Capa 1
</div>
<div id="capa2" style="position:absolute; width:100px; height:100px;
top:100px; left:200px; background-color:red; visibility:hidden">Capa 2
</div>
</body>
</html>

```

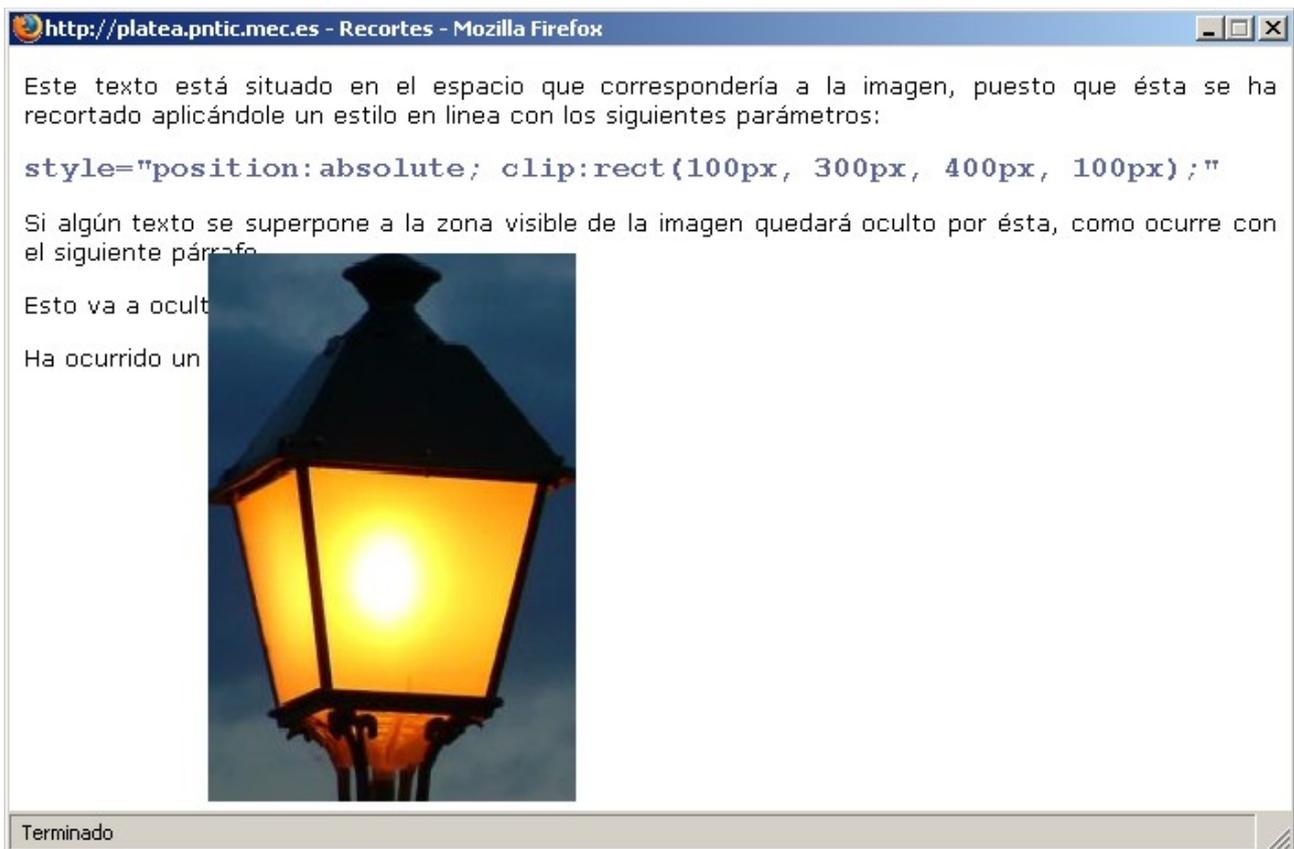


RECORTES

Tenemos la posibilidad de "encerrar" un elemento dentro de un recuadro con unas dimensiones determinadas. El resto del contenido quedará recortado y no se mostrará. Esta propiedad no está incluida en el editor CSS de N|VU por lo que habrá que insertar el código manualmente.

Propiedad: `clip`

Valores: Para definir la zona de recorte estableceremos las coordenadas de un rectángulo que hará de ventana visible del documento. Los valores que se aplican indican la posición de los vértices superior izquierdo, superior derecho, inferior derecho e inferior izquierdo respecto a los bordes superior e izquierdo del contenedor en el que se encuentra la imagen, y se anotan en la forma `rect(valor, valor, valor, valor)` donde el primer valor sirve para calcular la distancia hasta el borde superior, el segundo la distancia de ese vértice hasta el borde izquierdo, el tercero se calcula de nuevo respecto al borde superior y el último se calcula respecto al borde izquierdo y sirve para determinar la distancia del último vertice (y también del primero del que sólo conocíamos su distancia al borde superior) **Ejemplo:**



Internet Explorer 6 no soporta la propiedad `clip`.

DESBORDAMIENTO

¿Qué ocurre cuando definimos unas dimensiones exactas para un elemento y el contenido del mismo desborda los límites establecidos? Dependerá de los valores que adopte la propiedad que controla el desbordamiento. En el editor de CSS de N|VU podrás acceder a esta propiedad dentro de la pestaña caja.

Propiedad: `overflow`

Valores: Si no se especifica valor de `overflow` los navegadores utilizarán `visible` como valor por defecto. Los valores posibles son:

- **Visible:** Se desprecia la altura establecida, ampliándola hasta el tamaño necesario para que se pueda mostrar todo el contenido. Hay una diferencia importante en el modo de realizar este ajuste automático de la altura: Internet Explorer desplaza hacia abajo el resto de los elementos, mientras que Mozilla, Konqueror o Galeón hacen la ampliación pero mantienen la referencia de la altura original para mostrar los siguientes elementos, con lo que el resultado es que el contenido del elemento que se autoamplia se superpone al de los siguientes.
- **Hidden:** Se recorta el contenido mostrando lo que cabe en las dimensiones establecidas, pero no se muestran barras de desplazamiento para poder acceder al resto del contenido
- **Scroll:** Se incorporan al bloque barras de desplazamiento horizontal y vertical, incluso si no son necesarias porque el contenido ocupe menos espacio que el establecido en las dimensiones del elemento. Igual que en el caso anterior se recorta el contenido, pero ahora sí que podremos visualizarlo al disponer de barras de desplazamiento.

- **Auto:** Si el contenido no puede mostrarse completo en las dimensiones marcadas se incorpora una barra de desplazamiento vertical para poder acceder a lo que no queda visible. Si existe algún elemento que desborde las dimensiones horizontales también se incluiría una barra de desplazamiento horizontal.

Ejemplo:

```

<style type="text/css">
<!--
#auto {
overflow: auto;
}
#visible {
overflow: visible;
}
#oculto {
overflow: hidden;
}
#barras {
overflow: scroll;
}
-->
</style>

```

Visualizar el formato resultante

Esta capa tiene establecida la propiedad **overflow** en **auto** y se muestra completa aunque sus

Esta capa tiene establecida la propiedad **overflow** en **visible** y se muestra completa, invadiendo el espacio de los elementos que van a continuación y superponiéndose a ellos.

Esta capa tiene establecida la propiedad **overflow** en **scroll** y se muestra

⚠ Si has visto la página con Internet Explorer habrás comprobado que con el valor **overflow: visible** la capa aumenta su tamaño y desplaza al resto de elementos en lugar de superponerse a ellos.

Terminado

ALINEAMIENTO VERTICAL

Es una propiedad que tiene su principal aplicación para las imágenes, cosa que ya se podía hacer con HTML, pero puede aplicarse a cualquier elemento en línea. Esta propiedad no está incluida en el editor CSS de N|VU por lo que habrá que insertar el código manualmente.

Propiedad: `vertical-align`

Valores y ejemplos:

Se utiliza la imagen  para mostrar el efecto de cada valor

- **baseline** el elemento afectado alinea su borde inferior con el borde inferior del elemento padre. 
- **sub** el elemento afectado alinea su borde inferior como si fuera un subíndice 
- **super** el elemento afectado alinea su borde inferior como si fuera un superíndice 
- **top** el elemento afectado alinea su borde superior con el borde superior del elemento más alto de la línea, independientemente de que sea **texto** o imagen  
- **text-top** el elemento afectado alinea su borde superior con el borde superior de la fuente base del elemento padre (nótese que no toma en cuenta las modificaciones del **tamaño**  de la fuente)
- **middle** el elemento afectado alinea su línea central con con punto central del elemento padre 
- **bottom** el elemento afectado alinea su borde inferior con el elemento que se encuentra más **bajo** de la línea  (en este caso se toma en cuenta la modificación de tamaño de la fuente que ha desplazado el borde inferior de la letra j)
- **text-bottom** el elemento afectado alinea su borde inferior con el borde inferior de la fuente base del elemento **padre**  (nótese que no toma en cuenta el elemento más bajo)
- **longitud** expresada en cualquiera de las unidades válidas, tomando como referencia el borde inferior del elemento afectado e indicadas en positivo (.3cm)  o en negativo (-10px) 
- **porcentaje** donde la posición 0%  equivaldría a **baseline**, pudiendo expresarlo en negativo (-50%)  o positivo (100%) 

APARIENCIA DEL CURSOR

El navegador modifica de forma automática la forma en que muestra el cursor en función de la posición del ratón sobre los elementos de la página, pero también podemos especificarlo mediante estilos. Esta propiedad no está incluida en el editor CSS de N|VU por lo que habrá que insertar el código manualmente.

Propiedad: `cursor`

Valores y ejemplos: (El efecto sólo es perceptible en el documento HTML) Pasando el ratón sobre cada uno de los valores verás cambiar el cursor de acuerdo a cada uno de ellos: `crosshair, default, pointer, move, e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize, text, wait, help`.

APILAR CAPAS

Además de poder colocar las capas en la posición que se desee, también se puede decidir el orden en el que se irán apilando, de forma que unas zonas se superpongan a otras. Esta propiedad sólo puede aplicarse a capas que tengan establecido el atributo position. En el editor de CSS de NJVU podrás acceder a esta propiedad dentro de la pestaña caja.

Propiedad: `z-index`

Valor: Se aplican valores numéricos, de forma que un valor más alto indica que el elemento está por encima de los demás que tengan valor más bajo o que no tengan establecida propiedad de profundidad

Ejemplo:



- 94** • Experimenta con las propiedades para el diseño visual con CSS e incorpora aquellas que necesites a tus documentos o a tu hoja de estilo.

ABSOLUTAMENTE RELATIVO

Este título aparentemente paradójico procede el título original de un artículo en inglés *Relatively Absolute* de Tommy Olson en el que se intenta ofrecer una aclaración del sentido de los diversos modos de posicionamiento mediante CSS. De hecho el contenido de este apartado del curso no es una traducción del mismo pero sí una adaptación basada en lo que allí se plantea.

Para diseñar una página web, además de las decisiones referentes a las fuentes, colores y elementos gráficos, hay que elegir una disposición de los componentes para lo cual se pueden emplear tablas, flotaciones y posicionamiento.

El uso de las tablas, aunque sigue siendo probablemente el método más extendido es el menos recomendable de todos por su incidencia negativa en la accesibilidad y su falta de adecuación para una correcta estructuración de la información. De hecho está totalmente desaconsejado con el W3C.

La utilización de la propiedad float es un método mucho más acorde con los estándares y puede ser suficiente en muchas ocasiones.

El posicionamiento es la tercera técnica y la que más dificultades de interpretación tiene, por lo que vamos a intentar aclararlo.

Para empezar vamos a tomar como referencia una serie numérica: 1, 2, 3,... en la que cada número ocupa de forma natural una posición: el 2 aparece detrás del 1 y antes del 3. Los elementos que componen una página web siguen el mismo esquema y tienen un orden natural que es su orden de aparición en el código que genera la página, por lo que podríamos decir que coincidirían con el comportamiento de una serie numérica. (Fíjate que he especificado el orden de aparición en el código, no el orden de aparición en la página porque podríamos confundirnos con lo que vemos si nos guiamos por la presentación de los elementos en pantalla)

Una vez fijada la referencia consideremos los diferentes valores que puede adoptar la propiedad `position`

static

Es el valor por defecto. Cualquier elemento con `position:static` se situará en la ventana en la posición por defecto de acuerdo a su orden de aparición en el flujo del documento, sin modificar la posición. Dada esta particularidad podría pensarse que su efecto será nulo. En parte es así ya que no sirve para modificar la posición de un elemento, pero en algún momento será necesario recurrir a este valor para restaurar los valores por defecto.

relative

Algunas veces el lenguaje nos juega malas pasadas y esta puede ser una de ellas. La idea de posición relativa nos sugiere que debemos considerar un elemento ajeno como punto de referencia para situar elemento al que le aplicamos esta propiedad y realmente no es así: la posición relativa toma como referencia la posición que ocuparía el elemento si se respeta el flujo del documento sin modificar la situación que le correspondería en ese caso.

Si le aplicamos a un elemento la propiedad `position:relative` pero no especificamos ningún valor de desplazamiento el efecto es el mismo que si no la hubiéramos aplicado o hubiéramos utilizado el valor por defecto `position:static`. Si le adjudicamos un valor `top:10px` el elemento se desplazará 10 píxeles hacia abajo respecto a la que habría sido su posición normal. También podríamos ponerle valores negativos. Además, dado que lo único que hacemos es desplazar el elemento pero éste sigue ocupando lo mismo, es innecesario especificar un valor para `bottom` o `right` si lo hemos hecho para `top` y `left` (o viceversa)

Los elementos posicionados relativamente siguen en el flujo del documento ocupando el espacio que les corresponde, pero no donde aparecen sino donde habrían estado si hubieran estado posicionados estáticamente. Por ello la utilidad del posicionamiento relativo no se aplica excesivamente al desplazamiento de elementos, sino más bien a convertir elementos en posicionados, ya que una vez aplicada, aunque no movamos el elemento ni un píxel servirá como referencia para que los elementos hijos, que hereden alguna propiedad del que tiene `position:relative`, dispongan de un nuevo punto de cálculo, especialmente cuando se utilice el valor absoluto.

absolute

Si lo anterior puede habernos procucido un cierto desconcierto, esto va a acabar de complicar la situación: la interpretación de posición absoluta que podríamos deducir del término tampoco es correcta.

Cuando adjudicamos a un elemento `position:absolute` y le aplicamos un desplazamiento la nueva posición se calcula localizando el primer ancestro posicionado y situando los bordes a la distancia indicada respecto a los bordes de ese elemento. En caso de que no exista ningún ancestro posicionado la especificación CSS2. toma como referencia el bloque contenedor inicial, indicando que será el navegador el que determine cuál es, lo cual suele traducirse en que se toma como referencia el elemento `body` que coincide con la ventana de la aplicación.

Una cuestión que hay que tener en cuenta es que, cuando un elemento está posicionado con esta propiedad, sale del flujo del documento, por lo que el lugar que le correspondería es ocupado por el siguiente elemento, lo cual supone que podemos encontrarnos con que un elemento puede quedar debajo de otro que esté situado de forma absoluta.

En los navegadores con soporte completo para CSS se pueden especificar valores para los cuatro costados y el navegador se encarga de calcular el ancho, pero no ocurre así con Internet Explorer, lo cual nos obliga a especificar, al menos, la anchura del elemento posicionado. Se pueden indicar valores negativos para las distancias de los cuatro lados, aunque siempre habrá que tener en cuenta cuál es el bloque contenedor de referencia para evitar que nuestro elemento quede fuera de la pantalla.

fixed

En este caso el elemento de referencia es siempre la ventana del navegador, por lo que sería la posición más cercana a lo que podríamos interpretar como absoluto. Los elementos que llevan esta posición salen del flujo y se mantienen fijos en la ventana, pero aún mantienen una relación de profundidad con el resto de los elementos: mientras no hayamos alterado el valor z-index, los elementos que se han creado antes del posicionado se encontrarán por debajo de él, mientras que los que se crean después se superpondrán al elemento posicionado en el caso de que tengan que coincidir al desplazar la página.

Como consecuencia de lo anterior, si queremos crear un menú que permanezca fijo en la parte superior de la pantalla y creamos el elemento que lo contiene en primer lugar tendremos que modificar su valor z-index poniéndolo mayor que cualquier otro en el resto de la página.