

Curso Web accesible con XHTML y CSS.

Bloque V. Efectos visuales

Cursos Thales CICA Web 2006.
Curso WEBCSS

A. Gámez, L.M. Marín, F. Mesa & S. Fandiño

Efectos visuales

Índice

1. Introducción al formato visual
2. La propiedad display.
3. Determinando la posición de la caja
4. Uso de capas: la propiedad z-index
5. Propiedades float y clear
6. Determinando el tamaño de los elementos
7. Efectos de desbordamiento, recorte y visibilidad.

Efectos visuales

1. Introducción al formato visual

Hemos visto hasta ahora algunas propiedades (bordes, márgenes, rellenos) que pueden aplicarse a las cajas. Ahora vamos a ver cómo se puede definir el tipo y dimensiones de esas cajas, su comportamiento y relación con las otras cajas en la estructura del documento. El modelo de formato visual rige el comportamiento de las cajas generadas por los elementos de la página. Desde el punto de vista de una hoja de estilos, todo elemento html, y los elementos que lo componen son cajas. El modelo de caja de CSS describe las cajas rectangulares que son generadas por los elementos. Cada elemento genera una caja que actúa como bloque de contención de las cajas generadas subsecuentemente.

En HTML existen tres tipos de elementos en cuanto al modo en que son mostrados:

Elementos de bloque

Son aquellos tratados visualmente como bloques separados de los elementos que lo rodean (por ejemplo: `<p>` o `<div>`). Para hacerlo más sencillo, podemos decir que son aquellos que comienzan una nueva línea dentro del documento. Los elementos a nivel de bloque generan una *caja de bloque principal* que sólo contiene otras *cajas de bloque*.

Elementos de línea

Son aquellos que no forman nuevos bloques de contenido; el contenido es distribuido a nivel de las líneas (por ejemplo: ``, ``) o ``).

Elementos de lista

Son elementos de bloque que generan una caja principal y otras cajas adicionales (generalmente contienen una viñeta o caracteres alfanuméricos) que se agregan al costado del elemento. Ejemplos de estos elementos son `&ul ; b>`, ``) o `<dl>`)

Sin embargo, aunque cada elemento pertenece a uno de esos tres tipos, podemos alterar su comportamiento y hacer que actúe como si perteneciera a otra usando la propiedad `display`. Esta propiedad se verá con detalle más adelante, pero podemos adelantar algunos ejemplos:

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

Hemos alterado el comportamiento visual de modo que un elemento a nivel de línea en HTML (en este caso ``) se transforme en un elemento de bloque, de modo que cualquier texto marcado con `` comenzaría en una nueva línea. De igual modo, podríamos hacer que cualquier elemento se comporte como una tabla o como parte de una tabla.

Aunque las hojas de estilo lo permiten, no parece muy recomendable cambiar las características del tipo de elemento usando la propiedad `display`. Siempre pueden usarse nuevos elementos de bloque usando clases del elemento `<div>` o nuevos elementos de línea usando clases del elemento ``.

Con el formato visual vamos además a poder definir de cada elemento otras características como la posición, el desplazamiento, la superposición, etc.

2. La propiedad display.

Como hemos comentado, la propiedad display permite cambiar el tipo en que son mostrados los elementos. Su sintaxis es la siguiente:

display

Sintaxis: display: <valor>

Valores posibles: inline | block | list-item | run-in | compact | marker | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | none | inherit

Valor inicial: El que tenga por defecto cada elemento.

Se aplica a: Todos los elementos

Heredado: No

Vemos el significado de alguno de estos valores:

block

Este valor provoca que un elemento genere una caja de bloque principal

inline

Este valor provoca que un elemento genere una o más cajas de línea

list-item

Este valor provoca que un elemento genere una caja de bloque principal y una caja de línea list-item (por detalles ver la sección sobre listas)

marker

Este valor declara que el contenido generado antes o después de una caja será un marcador (por detalles ver la sección sobre marcadores)

none

Este valor provoca que un elemento no genere ninguna caja (es decir, el elemento no tiene ningún efecto sobre la composición) y los elementos descendientes tampoco generan cajas. Hay que destacar que este valor no crea una caja invisible sino que hace que el elemento desaparezca por completo. Esa es la diferencia con las propiedades sobre visibilidad que provocan que un elemento pueda ser invisible pero siga ocupando un espacio en la página

run-in y compact

Estos valores crean cajas de bloque o de línea según el contexto y tienen un comportamiento como el que conocemos para las listas de definiciones

table, inline-table, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell y table-caption

Estos valores provocan que un elemento se comporte como un elemento de una tabla. Usándolos podemos conseguir que cualquier elemento se comporte como una tabla o como parte de una tabla. Con HTML no tiene mucho sentido ya que existen elementos como las tablas, columnas y celdas, pero otros lenguajes (XML, por ejemplo) no tienen dichos elementos.

Aunque CSS permita cambiar el tipo de un elemento, los tipos por defecto (bien tipo línea, bien tipo bloque o elemento de lista) suelen ser bastante apropiados. Los cambios de esta propiedad pueden tener efectos indeseados, por lo que debe usarse con precaución.

3. Determinando la posición de la caja

Usando las propiedades que veremos a continuación podemos fijar la posición del elemento en la página.

El margen se divide en cuatro segmentos: top (superior), bottom (inferior), left (izquierdo) y right (derecho). De este modo podemos distinguir margin-left, margin-right, margin-top y margin-bottom.

La propiedad position

Sintaxis: position: <valor>

Valores posibles: static | relative | absolute | fixed | inherit

Valor inicial: static

Se aplica a: Todos los elementos

Heredado: No

La posición de la caja podrá desplazarse respecto de la que normalmente ocuparía usando las propiedades 'top', 'right', 'bottom' y 'left' que se verán a continuación. Esta propiedad position determina como actuarán dichas propiedades:

static

Esta es la posición predeterminada y la que conocemos en HTML. La caja se sitúa dentro del flujo normal de la página y las propiedades 'top', 'right', 'bottom' y 'left' no se aplican.

relative

La posición de la caja se ajusta en relación a su posición normal dentro de la página. Cuando una caja X se posiciona relativamente la caja siguiente se sitúa como si X no se hubiera desplazado.

absolute

Las cajas son quitadas del flujo normal de la página y su posición se especifica con las propiedades 'left', 'right', 'top', y 'bottom'. Estas propiedades especifican los desplazamientos con respecto al bloque de contención de la caja por lo que los elementos posicionados absolutamente no tienen ninguna influencia sobre la posición de las cajas siguientes. El bloque de contención para una caja posicionada es establecido por el antepasado posicionado más cercano o, si no existe, por el bloque de contención inicial (la esquina superior izquierda de la página, en el modelo visual).

fixed

El posicionamiento fijo es una subcategoría del posicionamiento absoluto. La única diferencia es que para una caja posicionada de modo fijo, el bloque de contención es establecido por el acceso visual (la pantalla del monitor) y el

elemento no se mueve cuando se realiza un desplazamiento. Esto significa que cuando se hace un scroll en la página los elementos con `position: fixed` mantienen su posición en la pantalla.

Las propiedades 'top', 'right', 'bottom' y 'left' tienen la siguiente sintaxis:

Propiedades top, right, bottom y left.

Sintaxis: top: <valor>
right: <valor>
bottom: <valor>
left: <valor>

Valores posibles: medida | porcentaje | auto | inherit

Valor inicial: auto

Se aplica a: Todos los elementos

Heredado: No

Estas propiedades determinan la posición del borde correspondiente del elemento. Los valores para las cuatro propiedades tienen los siguientes significados:

medida

El desplazamiento es una distancia fija desde el borde de referencia.

porcentaje

El desplazamiento es un porcentaje del ancho (para 'left' o 'right') o de la altura (para 'top' y 'bottom') del bloque de contención. Para 'top' y 'bottom', si la altura del bloque de contención no está especificada explícitamente (es decir, depende de la altura del contenido), el valor del porcentaje es interpretado como 'auto'.

auto

El efecto de este valor varía de acuerdo a otras propiedades (como el tamaño).

Para las cajas con `position: absolute`, el desplazamiento es con respecto al bloque de contención de la caja. Para las cajas con `position: relative`, el desplazamiento es con respecto al borde externo de la propia caja (es decir, la caja de desplaza de su posición normal dentro de la página de acuerdo a estas propiedades). Para las cajas con `position: fixed` el bloque de contención es establecido por el acceso visual (la pantalla del monitor) y el elemento no se mueve cuando se realiza un desplazamiento. Esto significa que cuando se hace un scroll en la página los elementos con `position: fixed` mantienen su posición en la pantalla.

4. Uso de capas: la propiedad z-index

Al poder ubicar los elementos en posiciones precisas de la página, puede ocurrir que dos elementos se superpongan. En los casos en que dos elementos queden superpuestos, podemos determinar el orden en que se "apilarán" a través de la propiedad z-index.

Por lo general, el navegador usa el mismo orden que los elementos tienen en la página para definir el orden en que son apilados. Este orden dentro de la pila de elementos puede adjudicarse explícitamente con la propiedad z-index utilizando un valor entero (cuanto más alto el entero, más cercano al lector o más arriba en la pila).

Se dice que un elemento establece un *contexto de pila* al cual pertenecen todos sus descendientes. En cada uno de esos contextos se establece un nuevo nivel de pila para los elementos descendientes. Y aquí lo importante: el *contexto de pila* (¡no la propiedad z-index!) es heredado y los elementos pertenecientes a distintos contextos no pueden "mezclarse".

Veamos el significado de lo anterior con un ejemplo: si en una página encontramos dos elementos (**A** y **B**), uno con z-index=5 (**A**) y el otro con z-index=3 (**B**), **A** será procesado al frente de **B**. Supongamos ahora que **A** tiene un descendiente (**C**) con z-index=0: el elemento **C**, por pertenecer al contexto de pila del elemento ubicado más arriba, también será procesado por encima de **B** (aunque **B** tenga z-index=3). *Todos* los elementos descendientes de **A** se ubicarán por encima de **B** (y también por encima de todos los descendientes de **B**).

Estas reglas pueden resultar confusas pero son bastante intuitivas, ya que resulta lógica que existiendo dos elementos con distinto orden, *todos los descendientes del elemento ubicado más al frente* se ubiquen por sobre *todos los descendientes del elemento ubicado más atrás*. Por supuesto, la misma regla se aplica recursivamente a los descendientes de un mismo elemento (si dentro de **C** encontramos dos elementos (**D** y **E**), uno con..., y así infinitamente).

Cuando más potencia adquiere esta propiedad es al usar algún tipo de rutina javascript. Así es posible conseguir que al realizar una determinada acción (pasar el ratón sobre un área determinada o hacer click en un botón, por ejemplo) se cambien los valores de esta propiedad, con lo que se pueden conseguir efectos muy llamativos.

5. Propiedades float y clear

Se puede especificar que los elementos queden flotando a la izquierda o a la derecha de los que le rodean. Esto se consigue con 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 (se lo hace "flotar" hacia uno u otro costado).

La propiedad clear es complementaria 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.

Propiedad float.

Sintaxis:	float: <valor>
Valores posibles:	none left right both inherit
Valor inicial:	none
Se aplica a:	Todos los elementos menos los posicionados y los contenidos generados
Heredado:	No

Los valores de esta propiedad tienen los siguientes significados:

left

El elemento genera una caja de bloque que flota a la izquierda. El contenido fluye sobre el costado derecho de la caja, comenzando en la parte superior.

right

Igual que 'left', pero el contenido fluye sobre el costado izquierdo de la caja, comenzando en la parte superior.

none

La caja no es flotante.

Propiedad clear.

Sintaxis:	float: <valor>
Valores posibles:	none left right both inherit
Valor inicial:	none

Se aplica a: Los elementos a nivel de bloque

Heredado: No

Los valores tienen los siguientes significados:

left

El margen superior de la caja generada se aumenta lo suficiente para que su borde superior quede debajo del borde inferior de cualquier caja flotante a la izquierda que aparezca antes en el documento fuente.

right

El margen superior de la caja generada se aumenta lo suficiente para que su borde superior quede debajo del borde inferior de cualquier caja flotante a la derecha que aparezca antes en el documento fuente.

both

La caja generada se mueve debajo de todas las cajas flotantes que aparecen antes en el documento fuente.

none

No existe ninguna restricción a la posición de la caja con respecto a los flotantes.

6. Determinando el tamaño de los elementos

Existen determinadas propiedades para fijar el tamaño de un determinado elemento. Son las siguientes:

Ancho del contenido: propiedad width

Sintaxis: width: <valor>

Valores posibles: medida | porcentaje | auto | inherit

Valor inicial: auto

Se aplica a: Todos los elementos excepto los elementos de línea no reemplazados, las filas de las tablas y los grupos de filas

Heredado: No

Los valores de esta propiedad tienen los siguientes significados:

medida

Especifica un ancho fijo.

porcentaje

Especifica el ancho según un porcentaje. El porcentaje es calculado con respecto al ancho del bloque de contención de la caja.

auto

El ancho depende de los valores de otras propiedades.

Altura del contenido: propiedad height

Sintaxis: height: <valor>

Valores posibles: medida | porcentaje | auto | inherit

Valor inicial: auto

Se aplica a: Todos los elementos excepto los elementos de línea no reemplazados, las columnas de las tablas y los grupos de columnas

Heredado: No

Los valores de esta propiedad tienen los siguientes significados:

medida

Especifica un ancho fijo.

porcentaje

Especifica el ancho según un porcentaje. El porcentaje es calculado con respecto al ancho del bloque de contención de la caja.

auto

El ancho depende de los valores de otras propiedades.

Alturas y anchuras mínimas y máximas.

También es posible determinar un valor mínimo o máximo que ocupará un determinado elemento. Esto se consigue usando las propiedades min-width, max-width, min-height y max-height. Los valores se pueden indicar de modo absoluto, indicando una medida, o relativo, indicando un porcentaje que será calculado respecto al bloque de contención.

7. Efectos de desbordamiento, recorte y visibilidad.

Existen propiedades que permiten aplicar algunos efectos visuales en la página: recortar una parte de la caja de un elemento, especificar el comportamiento para los casos en que el contenido de un elemento es mayor que su tamaño y provocar que un elemento sea invisible.

Generalmente el contenido de una caja de bloque se mantiene dentro de los límites de la caja, pero puede suceder que el contenido desborde esos límites y quede parcial o totalmente fuera de la caja.

Esto puede suceder si especificamos un tamaño determinado para un elemento (con las propiedades 'width' y 'height', por ejemplo) y su contenido resulta demasiado grande para las medidas adjudicadas. También puede darse al utilizar márgenes negativos o posiciones absolutas para el elemento.

Cuando se produce un desbordamiento, la propiedad 'overflow' especifica si la caja es recortada y (en caso afirmativo) cómo será recortada. La propiedad 'clip' especifica el tamaño y la forma de la zona de recorte.

Propiedad overflow.

Sintaxis: overflow: <valor>

Valores posibles: visible | hidden | scroll | auto | inherit

Valor inicial: : visible

Se aplica a: Los elementos a nivel de bloque y reemplazados

Heredado: No

Esta propiedad especifica si el contenido de un elemento a nivel de bloque es recortado cuando desborda la caja del elemento. Los valores tienen los siguientes significados:

visible

Este valor indica que el contenido no es recortado, es decir, puede ser procesado fuera de la caja de bloque.

hidden

Este valor indica que el contenido es recortado y los usuarios no tendrán acceso al contenido recortado. El tamaño y forma de la zona de recorte son especificados por la propiedad 'clip'.

scroll

Este valor indica que el contenido es recortado y el navegador debe proporcionar un mecanismo de desplazamiento que permanecerá siempre visible (aunque la caja no tenga parte de su contenido recortado).

auto

El comportamiento del valor 'auto' depende del navegador, pero significa que éste debe proporcionar un mecanismo de desplazamiento para las cajas desbordadas.

Propiedad clip.

Sintaxis: clip: <valor>

Valores posibles: <forma> | auto | inherit

Valor inicial: auto

Se aplica a: Los elementos a nivel de bloque y reemplazados

Heredado: No

Una zona de recorte define qué porción del contenido de un elemento es visible. De manera predeterminada, la zona de recorte tiene el mismo tamaño y forma que la caja del elemento. Sin embargo, la zona de recorte puede ser modificada por la propiedad 'clip'.

La propiedad 'clip' se aplica a elementos que tienen una propiedad 'overflow' con un valor diferente a 'visible'. Los valores tienen los siguientes significados:

auto

La zona de recorte tiene el mismo tamaño y ubicación que la(s) caja(s) del elemento.

<forma>

En CSS2, el único valor permitido para es: rect (<arriba>, <derecha>, <abajo>, <izquierda>).

Los valores para <arriba><derecha><abajo> e <izquierda> pueden ser una <medida> o 'auto'. Las medidas negativas están permitidas y 'auto' significa lo mismo que '0' (cero).

Los profesores de este Curso WebCSS son:

A. Gamez, L.M. Marin, F. Mesa & S. Fandiño