



UD02. CSS. Modelo Caja

Diseño de interfaces WEB
2º Curso

- 
- block & inline
 - display
 - box-sizing
 - overflow
 - márgenes
 - Propiedades caja CSS3

display

block vs inline

Todos los elementos HTML disponen de una propiedad *display*

Indica como los elementos van a ser mostrados

Casi todos los elementos tiene la propiedad con valor *block* o *inline*

block:

Empieza en una nueva línea y ocupa el ancho disponible. “En principio” sólo puede tener hijos *inline* (aunque algunos *block* admiten otros *block*, aunque no todos a todos)

`<h1>` `<p>` `<div>`

inline:

No empieza en una nueva línea y ocupa sólo el ancho necesario. “En principio” sólo puede tener hijos *inline*.

`` `` `<a>`

display

display:none vs visibility:hidden

Ambos dos valores ocultan el elemento

none:

El elemento “desaparece” de la estructura. No sólo no se ve, sino que no afecta al resto de elementos

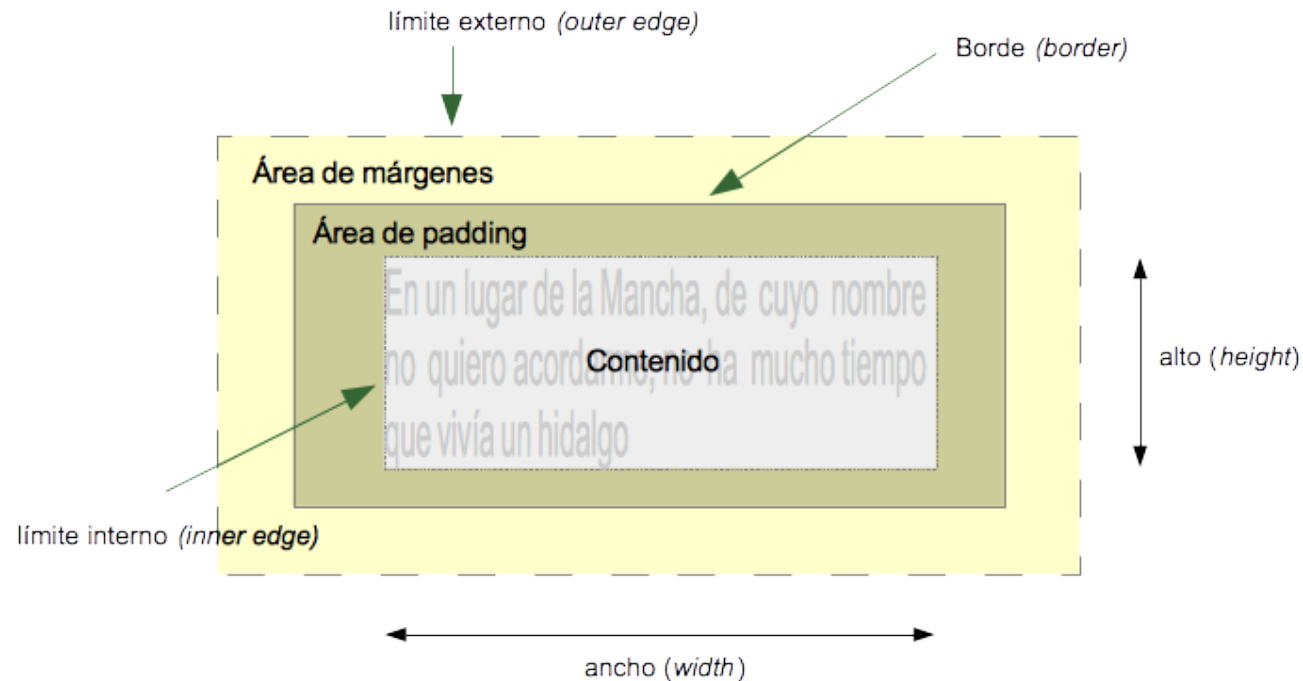
hidden:

El elemento simplemente se oculta. El *layout* del documento no cambia: todos los elementos se mantienen ubicados igual que si el elemento fuera visible.

modelo caja

Partes de la caja

Todos los elementos generan un rectángulo (una caja) que los envuelve



Algunas propiedades se comportan de manera diferente en función de si el elemento es el *block* o *inline*.

box-sizing

El área total que ocupa el elemento:

$(width + padding\ horz + margin\ horz) * (height + padding\ vert + margin\ vert)$

Es decir, *width* y *height* SOLO hacen referencia a contenido

A partir de CSS3 es posible indicar a que hace referencia, o al contenido o al contenido más el padding

Default value:	content-box
Inherited:	no
Animatable:	no. Read about animatable
Version:	CSS3
JavaScript syntax:	<code>object.style.boxSizing="border-box"</code>

content-box | border-box | initial | inherit

width & height

Default value:	auto
Inherited:	no
Animatable:	yes. Read about animatable
Version:	CSS1
JavaScript syntax:	<code>object.style.width="500px"</code>

Default value:	auto
Inherited:	no
Animatable:	yes. Read about animatable
Version:	CSS1
JavaScript syntax:	<code>object.style.height="500px"</code>

No afectan a elementos *inline*

longitud | porcentaje | *auto* / *inherit*

auto: el propio navegador el que calcula el tamaño (tan ancho como el navegador o como el elemento que lo contenga).

max/min width & height

max-width, min-width
max-height, min-height

Pensadas para definir los límites de los elementos.

Sólo funcionan en los *block* y elementos que son reemplazados (como *img*)

Sólo afectan al contenido

Prioridades

min-width > max-width > width

min-height > max-height > height

overflow

Lo más habitual es no definir la altura de los elementos y que esta sea calculada en función del tamaño del texto (así si el usuario cambia el tamaño, todo se debería adaptar)

Default value:	visible
Inherited:	no
Animatable:	no. Read about animatable
Version:	CSS2
JavaScript syntax:	<code>object.style.overflow="scroll"</code>

Si definimos una altura hay que estudiar que pasa si el contenido no cabe, si desborda (*overflow*)

Sólo funcionan en los *block* y elementos que son reemplazados (como *img*)

visible | *hidden* | *scroll* | *auto*

auto: el navegador decide

Márgenes

margin

Default value:	0
Inherited:	no
Animatable:	yes, see <i>individual properties</i> . Read about animatable
Version:	CSS1
JavaScript syntax:	<code>object.style.margin="100px 50px"</code>

Define el tamaño del *margin*

longitud | *auto* | *initial* | *inherit*

Admite 4 formatos: T R B L, T RL B, TB RL, all

La *longitud* se puede expresar como % del ancho del padre

auto: el navegador decide la cantidad de margen necesario para adaptarse o rellenar el espacio disponible.

Fuente: <http://www.w3schools.com/>

Márgenes

Márgenes por defecto

El navegador puede definir márgenes alrededor de encabezados, párrafos, etc.

Es una buena práctica inicializar esos los valores a un valor conocido.

```
* {margin: 0px; padding: 0px; }
```

Márgenes

Colapso de márgenes

Si los márgenes inferior y superior de dos elementos se solapan, sólo se usa el más grande

Márgenes en elementos *inline*

Los márgenes superior o inferior no afectan, pero el derecho e izquierdo sí, aunque eso implique crear varias líneas

Márgenes negativos

Es posible definir márgenes negativos. El efecto que se crea es el de “atraer” a los elementos del lado correspondiente al que se haya aplicado el *margin*

Otras propiedades

padding

Default value:	0
Inherited:	no
Animatable:	yes, see <i>individual properties</i> . Read about animatable
Version:	CSS1
JavaScript syntax:	<code>object.style.padding="100px 20px"</code>

Define el tamaño del *padding*

longitud | % | *initial* | *inherit*

Admite 4 formatos: T R B L, T RL B, TB RL, all

La *longitud* se puede expresar como % del ancho del padre

Otras propiedades

border-width

Default value:	medium
Inherited:	no
Animatable:	yes. Read about animatable
Version:	CSS1
JavaScript syntax:	<code>object.style.borderWidth="1px 5px"</code>

Permite definir el grosor del borde

medium | *thin* | *thick* | longitud | *initial* | *inherit*;

Otras propiedades

border-color

Default value:	The current color of the element
Inherited:	no
Animatable:	yes. Read about animatable
Version:	CSS1
JavaScript syntax:	<code>object.style.borderColor="#FF0000 blue"</code>

Permite definir el color del borde

color | *transparent* | *initial* | *inherit*

Otras propiedades

border-style

Default value:	none
Inherited:	no
Animatable:	no. Read about animatable
Version:	CSS1
JavaScript syntax:	<code>object.style.borderStyle="dotted double"</code>

Permite definir el estilo del borde

none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | initial | inherit;

Otras propiedades

border

Default value:	medium none <i>color</i>
Inherited:	no
Animatable:	yes, see <i>individual properties</i> . Read about animatable
Version:	CSS1
JavaScript syntax:	<code>object.style.border="3px solid blue"</code>

Permite indicar de manera abreviada todas las propiedades de los bordes

border-width | border-style | border-color | initial | inherit

Otras propiedades

border-radius

Default value:	0
Inherited:	no
Animatable:	yes. Read about animatable
Version:	CSS3
JavaScript syntax:	<code>object.style.borderRadius="25px"</code>

Permite definir el radio de las esquinas de la caja

1-4 longitud | % | initial | inherit;

Admite 4 formatos: TL TR BR BL, TL TRBL BR, TLBR TRBL, all

Otras propiedades

box-shadow

Default value:	none
Inherited:	no
Animatable:	yes. Read about animatable
Version:	CSS3
JavaScript syntax:	<code>object.style.boxShadow="10px 20px 30px blue"</code>

Permite definir una sombra para la caja

none | h-shadow v-shadow blur spread color inset | initial | inherit;

Se admite una lista (separada por comas) de sombras, cada una definida por entre 2 y 5 valores:
posición H, posición V, blur, spread, color

inset sombra interna

Para aplicar sombras en textos: text-shadow

Fuente: <http://www.w3schools.com/>

Otras propiedades

Gradientes lineales

Se utiliza la función *linear-gradient*

Parámetros:

Punto inicial (o *top* | *bottom* | *left* | *right*) o ángulo del gradiente

Lista de colores separados por comas (dos como mínimo). Cada color puede tener asociado un % que indica cuando ha de finalizar

El ángulo se mide entre la horizontal y el vector director de gradiente

```
background: linear-gradient(top, #FFFFFF, #006699);
```

```
background: linear-gradient(30deg, #FFFFFF 50%, #006699 90%);
```

Otras propiedades

Gradientes radiales

Se utiliza la función *radial-gradient*

Parámetros:

Forma: *ellipse* | *circle*

Lista de colores separados por comas (dos como mínimo). Cada color tiene asociado un % que indica cuando ha de finalizar

```
background: radial-gradient(red 5%, yellow 15%, green 60%);
```

```
background: radial-gradient(circle, #FFFFFF 0%, #006699 200%);
```

Otras propiedades

Transparencia

1) Expresando el color del elemento con transparencia

`rgba(r,g,b,opacity)`

opacity : [0--1] 0 transparente, 1 opaco

2) Utilizando la propiedad opacity:

Default value:	1
Inherited:	no
Animatable:	yes, see <i>individual properties</i> . Read about animatable
Version:	CSS3
JavaScript syntax:	<code>object.style.opacity="0.5"</code>

numero | *initial* | *inherit*

Otras propiedades

border-image-slice

Es posible configurar imágenes que den forma a los bordes con propiedades como *border-image* o *border-image-repeat*.

Una muy interesante es *border-image-slice* que permite trozear una imagen de manera que cada trozo se coloque en un trozo de la caja

Default value:	100%
Inherited:	no
Animatable:	no. Read about animatable
Version:	CSS3
JavaScript syntax:	<code>object.style.borderImageSlice="30%"</code>

numero | % | *fill* | *initial* | *inherit*

Otras propiedades

border-image-slice

Se pueden utilizar 4 valores.

Cada uno de ellos indica como recortar la imagen en cada lado, empezando por arriba y siguiendo las agujas del reloj. Es posible especificarlo en formato acortado

```
border: 12px;  
border-style:solid;  
border-image-source: url( "mi_borde.png" );  
border-image-slice: 12;
```

También se puede utilizar el valor *fill*, que rellena también el centro de caja con el centro de la imagen

¿Preguntas?