



TALLER – DISEÑO SITIO WEB EMPRESA

Objetivo: Profundizar en los comandos y etiquetas utilizadas en HTML y combinarlos con `<style>` para darle estilos a las páginas web.

Unidades de medida absolutas y relativas

Las unidades de medida a las que se hará referencia a continuación, son las que se aplican a la anchura y altura de los elementos de HTML. No todos los elementos de HTML poseen los mismos atributos en cuanto a sus dimensiones. Por ejemplo la etiqueta **div** tiene los atributos `width` y `height`; pero la etiqueta **span** no los tiene. En los ejemplos tendremos en cuenta los elementos más usados sobre los que se aplican la anchura y altura, como párrafos, divisiones, tamaño de letra, imágenes, etc.

Se pueden dividir las unidades de medida en dos grupos: relativas y absolutas.

Las unidades relativas se definen en función de otro elemento, como el ancho de la pantalla o el ancho del tipo de letra. Por ejemplo, si escribimos un `width: 40%`, queremos decir que ese elemento tendrá el 40% del ancho de la pantalla en la que se visualice (o el ancho de su contenedor). En realidad en este contexto cuando decimos pantalla, nos referimos a la ventana de visualización. Las unidades relativas son muy utilizadas en los diseños adaptables.

Las unidades absolutas son fijas y no cambian, se definen en si mismas respecto a alguna unidad de medida. Por ejemplo si escribimos un `width: 200px`, el elemento tendrá 200 pixels en cualquier pantalla. No hay que escribir un espacio en blanco entre el número y la unidad, `width: 200 px` sería erróneo; ni poner un punto al final, `width: 200px.` también estaría mal.

Unidades absolutas

PX, El píxel es la unidad más utilizada hasta la fecha. Por ejemplo, 200px. En algunos casos, como en los márgenes puede ser un número negativo. En ocasiones también se puede utilizar el valor cero sin unidades, por ejemplo `margin: 0`, indica sin margen y no necesita unidades. $1 \text{ px} = 0.75 \text{ pt}$. También se define el píxel como $1 \text{ px} = 1 \text{ pulgada} / 96$, por lo tanto $2,54 \text{ cm} / 96 = 0,26 \text{ mm}$. es decir, $1 \text{ px} = 0,26 \text{ mm}$. Nota: una pulgada son 2,54 cm.

El píxel es la unidad más utilizada en las dimensiones de las imágenes ya que en edición gráfica también se utiliza el píxel. También es habitual utilizarlo en las dimensiones de contenedores como **div**, así como en márgenes, **padding**. Hasta ahora también se utilizaba el píxel para definir el



tamaño de la fuente con font-size, pero se están utilizando cada vez más unidades relativas como em y rem.

PUNTOS, se escribe como pt, 1 pt = 1 pulgada/72 = 2,54/72=0,0352 cm. = 0,352 mm.

CENTÍMETROS, se escribe como cm. 1cm = 96 px/ 2.54

MILÍMETROS, se escribe como mm. 1mm = 1/10 cm

PULGADAS, se escribe como in, 1 in = 2.54 cm = 96 px

PICAS, se escribe pc, 1 pc = 1/6 in = 16 px.

Las unidades absolutas, como el píxel no cambian, si modificamos el tamaño de la ventana, puedes comprobarlo con este ejemplo donde mostramos un contenedor div de width:300px, debajo hay de otro contenedor div definido con una unidad variable como width:40%:

```
<div style="width: 300px; background-color:azure">Ancho fijo. Definido con width: 300px </div>
```

```
<br>
```

```
<div style="width: 40%; background-color:azure">Ancho variable. Definido con width: 40% </div>
```

Ancho fijo. Definido con width: 300px

Ancho variable. Definido con width: 40%

Colores

El sistema utilizado en las páginas web se basa en colores de 24 bits que permite 16,7 millones de colores por canal (256 x 256 x 256 = 16.777.216). Adicionalmente se puede añadir la transparencia/opacidad del color.

Los colores en CSS se pueden representar de varias formas: Nombres, RGB, palabras clave, colores del sistema, RGB hexadecimal, RGB numérico y RGB porcentual.

1. Nombres de colores.

Actualmente los navegadores soportan 140 nombres de color según la nomenclatura de X11/SVG.

Por ejemplo, color:green

2. Colores RGB.

Este sistema utiliza los colores primarios aditivos rojo (Red), verde (Green) y azul (Blue). Esto es conocido como el sistema RGB por las siglas de los colores en inglés.



De esta forma podemos crear un color indicando la cantidad de cada uno de los primarios que interviene en la mezcla. Para cada color podemos especificar 256 intensidades distintas, y al ser tres colores podemos obtener un total aproximado de 16,7 millones de colores.

En html, podemos definir un color RGB de varias formas:

- a) RGB con valores hexadecimales, siguiendo el formato #F7F0E2 (el más utilizado), con valores para cada color entre el 00 (mínimo) y FF (máximo).
- b) RGB con valores decimales, con la fórmula rgb(247, 240, 226), con valores entre 0 y 255.
- c) RGB con valores porcentuales, como rgb(90%, 60%, 35%).
- d) RGB con valores hexadecimales reducidos, si los dos valores de cada color son iguales, podemos reducirlos a uno, así, #FF00EE podemos escribirlo como #F0E.

Así, el color ROSADO lo podemos definir como #FA75C4, rgb(250, 117, 196) o rgb(98%, 46%, 77%).

3. Colores RGBA.

Si añadimos un valor de transparencia alpha al RGB, tenemos el RGBA. El valor 0 indica transparencia total, y el valor 1 indica transparencia nula.

Así, un color rgb(0, 111, 155) lo podemos definir con un 50% de transparencia de esta forma, rgba(0, 111, 155, 0.5) aunque se pueda confundir una mayor transparencia con un color más claro, en realidad lo que hace es dejar ver más el color de fondo.

4. Colores HSL

Los códigos de colores HSL se definieron en CSS en 2011, todavía no son muy utilizados pero son más intuitivos que los RGB ya que se definen mediante tres valores que podemos interpretar mejor que la mezcla de rojo, verde y azul:

- Matiz (Hue) es un número entre 0 y 360 y abarca todos los tonos de color.
- Saturación (Saturation) es un porcentaje que da la intensidad del color, 0% es poco intenso y 100% es más intenso.
- Luminosidad (Lightness) es el porcentaje que representa lo más o menos claro de un color, 0% es blanco y 100% es negro.

Así, un color hsl(120,50%,50%), que es un verde neutro, lo podemos hacer un 30% más intenso (saturado) con hsl(120,80%,50%) o un 30% más claro (luminoso) con hsl(120,50%,80%)



5. Colores HSLA

Los códigos de colores HSL también pueden tener transparencia añadiendo un número, 0 indica transparencia total, y el valor 1 indica transparencia nula.

Por ejemplo el color `hsla(120,50%,50%,1)` no tiene nada de transparencia, y el color `hsla(120,80%,50%, 0.4)` es un 40% transparente.

Atributo opacity

Anteriormente se ha hablado de la transparencia de un color, ahora se va a hablar del atributo `opacity`. Este atributo produce también un efecto de transparencia, pero no está incluido en la descripción del color, sino que se aplica como un atributo más a elementos de HTML como imágenes, contenedores, fondos, etc.

Por ejemplo, si lo aplicamos a una etiqueta `span`, el efecto sobre el color de fondo es el mismo que la transparencia de RGBA/HSLA:

```
<span style="background-color:rgb(0,111,155); opacity: 0.5">
```

Y si lo aplicamos a una imagen:

```

```

Además, hay una diferencia importante, si aplicamos el atributo `opacity`, la transparencia la heredarán todos los elementos hijos, cosa que no sucede con RGBA/HSLA.

Por ejemplo, en este primer caso la transparencia la hereda el texto que hay dentro del contenedor `div` para el que hemos fijado un valor de transparencia con `opacity:0.6`

```
<div style="background-color: rgb(0,111,155); opacity: 0.6; width: 30%">
```

Mientras, que en este segundo caso la transparencia **no** la hereda el texto que hay dentro del contenedor `div` para el que hemos fijado un valor de transparencia con RGBA 0.6.

```
<div style="background-color: rgba(0,111,155,0.6); width: 30%"><p>Con RGBA transparencia 0.6</p></div>
```

Si fijamos un valor de transparencia con `opacity:0`, el contenedor no se verá, aunque sigue ahí ocupando un espacio.

```
<div style="background-color: rgb(0,111,155); opacity: 0; width: 30%"><p>Con opacity:0</p></div>
```



Si lo que queremos es hacer desaparecer un contenedor u otro objeto disponemos de la propiedad `display:none` el contenedor no se verá y, además, no ocupará espacio.

```
<div style="background-color: rgb(0,111,155); display: none; width: 30%">    <p>Con  
display:none</p></div>
```

Para ocultar un contenedor u otro objeto sin alterar el diseño, es decir, dejando el hueco correspondiente, se suele utilizar la propiedad `visibility:hidden`, el contenedor no se verá pero ocupará su espacio.

```
<div style="background-color: rgb(0,111,155); visibility:hidden; width: 30%">    <p>Con  
visibility:hidden</p></div>
```

Atributo `visibility`

Como acabamos de ver el atributo `visibility` permite cambiar la visibilidad de un elemento.

Puede tomar los siguientes valores:

- visible, el elemento se ve, es el valor por defecto.
- hidden, el elemento no se ve, queda oculto pero ocupando espacio. Es decir, quedará un hueco.
- collapse, sólo se puede usar para tablas, se oculta el elemento sin dejar hueco y sin afectar a los bordes de la tabla.

La diferencia principal entre ocultar un elemento con `visibility` o con `opacity` es que con esta última los elementos hijos la heredan, mientras que con `visibility` los hijos no la heredan. Ambas dejan el hueco. Otra diferencia es que con `opacity` el hueco responde a los eventos, por ejemplo, podemos detectar un click sobre él, mientras que con `visibility` no.

Si no queremos dejar hueco, debemos usar la propiedad `display:none`. Puesto que no hay hueco, tampoco detecta los eventos

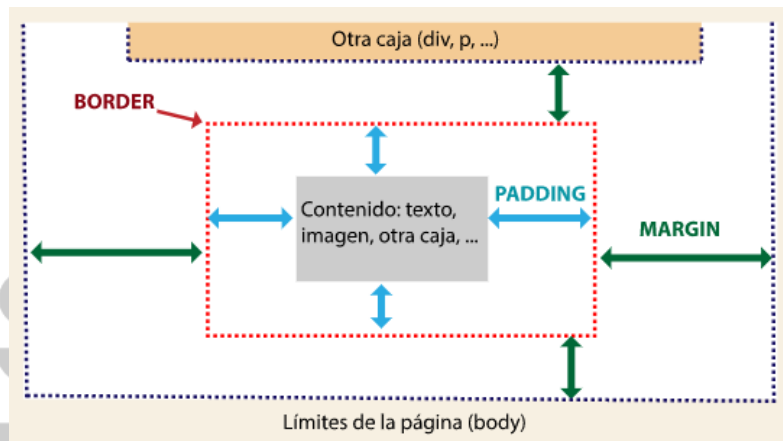
Modelo de caja

Los elementos de HTML están contenidos en cajas rectangulares. La etiqueta `body` es la caja principal dentro de la cual se van colocando las cajas de las etiquetas que se van creando, las cajas pueden estar unas dentro de otras, o unas al lado de otras, incluso se pueden superponer. Podemos pensar en el modelo de caja como un borde rectangular (`border`) que rodea al contenido, hay un



margen interior que define la propiedad padding y un margen exterior que define la propiedad margin. Todo esto es lo que se conoce como modelo de cajas de CSS (box model) y es fundamental entenderlo bien para poder diseñar páginas web con soltura.

Se va a explicarlo sobre este esquema:



Los componentes del modelo de caja son los siguientes:

a) Contenido. En cada caso este contenido es de un tipo, puede ser un texto, una imagen, una lista, otra caja, ... El contenido depende de cada etiqueta, por ejemplo, la etiqueta div admite varios tipos de contenido (texto, imágenes, otras etiquetas, ...) mientras que una etiqueta img sólo admite una imagen como contenido. En nuestro esquema el contenido está representado por el rectángulo gris.

b) Padding (margen interior). Espacio transparente entre el contenido y el borde. Puede ser de grosor cero. También podemos decir que el padding es el margen interior de la caja. En nuestro esquema el padding está representado por las dobles flechas azules.

c) Border (borde). Línea que rodea el padding, y por lo tanto también al contenido. Puede ser de grosor cero y por lo tanto invisible. En nuestro esquema el borde está representado por la línea punteada roja.

d) Margin (margen exterior). Espacio transparente rectangular que rodea al borde y sirve de separación con otros elementos adyacentes. Puede ser de grosor cero. En nuestro esquema el margin está representado por las doble flechas verdes.

Y aunque no son estrictamente del modelo de caja, también intervienen:

e) Background-image (Imagen de fondo). Algunos elementos como los contenedores div y los párrafos p pueden tener una imagen de fondo que se coloca por detrás del contenido, hasta el borde. En nuestro esquema no hay imagen de fondo. Más adelante veremos un esquema que sí la tiene.



f) Background-color (Color de fondo). Si se define un color de fondo, este se coloca por detrás de la imagen de fondo, hasta el borde. En nuestro esquema no hay color de fondo.

Por otro lado, un poco más adelante, veremos cómo afecta el modelo de caja a las propiedades width (anchura) y height (altura).

Valores por defecto.

Por defecto, los rectángulos del modelo de caja son invisibles, por eso los hemos dibujado con líneas de puntos en el esquema anterior. Es decir, el atributo border tiene inicialmente valor cero, es invisible. Y el rectángulo que rodea al contenido lo hemos dibujado para explicar el modelo pero no existe como tal, no se puede dibujar. Por supuesto, tampoco tienen valor por defecto el color de fondo y la imagen de fondo.

Sin embargo los valores por defecto del margen y del padding no son siempre cero. Algunas etiquetas tienen establecidos unos valores por defecto. Estos valores dependen de los navegadores web, aunque son similares en la mayoría de los casos. Por ejemplo, la etiqueta párrafo p tiene unos valores de 1em para los márgenes superior e inferior, y un valor cero para los márgenes derecho e izquierdo. Esto tiene sentido porque si no fuese así, dos párrafos quedarían pegados entre sí, no habría diferencia entre un salto de línea y un párrafo. También tienen valores por defecto las otras etiquetas de bloque (div, h1, h2, ...)

El padding suele no tener valor por defecto en la mayoría de etiquetas.

El padding es siempre transparente, y deja ver la imagen de fondo y el color de fondo de su propia caja.

El margin es siempre transparente, y deja ver la imagen de fondo y el color de fondo de la caja que lo contiene.

Margin

El atributo margin define el espacio transparente entre el borde de la caja y los elementos adyacentes. Es el margen exterior de la caja. Puede tener valores numéricos enteros positivos y negativos. El valor cero indica que no existe margen. Un valor negativo hace que se "introduzca" en el elemento contiguo. El margin se compone de cuatro valores, uno para cada lado. Se pueden indicar de forma compacta o de forma explícita.



- margin de forma explícita: margin superior con margin-top; margin derecho con margin-right; margin inferior con margin-bottom; margin izquierdo con margin-left.

- margin de forma compacta, en este caso hay cuatro variantes:

a) Con un sólo número. Se aplica el valor a las cuatro posiciones del margin. Por ejemplo margin:15px, es lo mismo que margin-top: 15px; margin-right: 15px; margin-bottom: 15px; margin-left: 15px.

b) Con dos números. Se aplica el primer valor al margin superior e inferior; y el segundo valor al margin derecho e izquierdo. Por ejemplo margin: 15px 25px, es lo mismo que margin-top: 15px; margin-right: 25px; margin-bottom: 15px; margin-left: 25px.

c) Con cuatro números. Se aplica el primer valor al margin superior, el segundo al derecho, el tercero al inferior y el cuarto al izquierdo. Por ejemplo margin:5px 15px 25px 35px, es lo mismo que margin-top: 5px; margin-right: 15px; margin-bottom: 25px; margin-left: 35px.

c) Con tres números. Se aplica el primer valor al margin superior, el segundo al derecho e izquierdo y el tercero al inferior. Por ejemplo margin:5px 15px 25px, es lo mismo que margin-top: 5px; margin-right: 15px; margin-left: 15px; margin-bottom: 25px.

Unidades.

El valor del margin se puede indicar con una unidad absoluta (px, mm, ..) o con una unidad relativa (% , em, ...). Se puede dar el valor auto que indica que se cree un margen de forma automática según los espacios disponibles con los elementos adyacentes, normalmente con los mismos valores arriba y abajo, y a derecha e izquierda; por esto se puede utilizar para centrar los elementos horizontalmente dando el valor auto a los márgenes izquierdo y derecho. También se puede heredar del elemento contenedor con el valor inherit.

El margin superior e inferior sólo es aplicable a elementos de bloque, como div y p, mientras que el margin izquierdo y derecho se pueden aplicar a todos los elementos.

Padding

El atributo padding define el espacio transparente entre el contenido y el borde. Es el margen interior de la caja, el margen que queda dentro de la caja. Puede tener valores numéricos enteros positivos. El valor cero indica que no existe padding. El padding se compone de cuatro valores, uno para cada lado. Igual que con el margin, el padding se puede indicar de forma compacta o de forma explícita.



Padding de forma explícita: Padding superior con padding-top; Padding derecho con padding-right; Padding inferior con padding-bottom; Padding izquierdo con padding-left.

Padding de forma compacta, en este caso hay cuatro variantes:

a) Con un sólo número. Se aplica el valor a las cuatro posiciones del padding. Por ejemplo padding:15px, es lo mismo que padding-top: 15px; padding-right: 15px; padding-bottom: 15px; padding-left: 15px. También hemos dado un margin: 10px.

b) Con dos números. Se aplica el primer valor al padding superior e inferior; y el segundo valor al padding derecho e izquierdo. Por ejemplo padding: 15px 25px, es lo mismo que padding-top: 15px; padding-right: 25px; padding-bottom: 15px; padding-left: 25px.

c) Con cuatro números. Se aplica el primer valor al padding superior, el segundo al derecho, el tercero al inferior y el cuarto al izquierdo. Por ejemplo padding:5px 15px 25px 35px, es lo mismo que padding-top: 5px; padding-right: 15px; padding-bottom: 25px; padding-left: 35px.

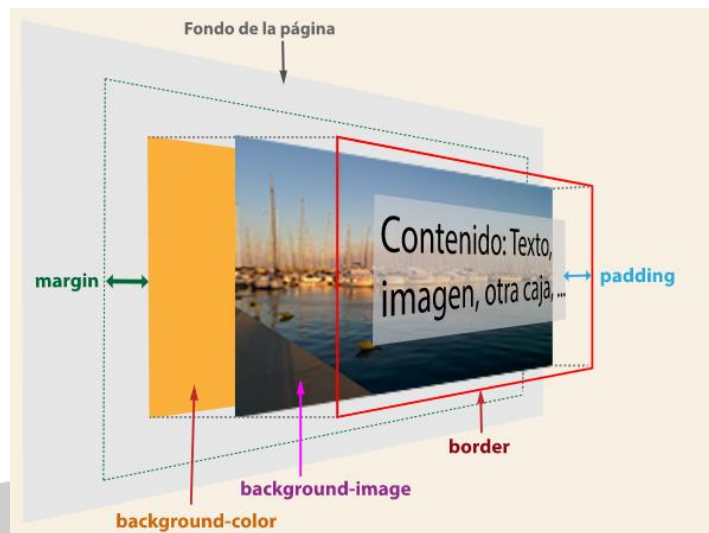
d) Con tres números. Se aplica el primer valor al padding superior, el segundo al derecho e izquierdo y el tercero al inferior. Por ejemplo padding:5px 15px 25px, es lo mismo que padding-top: 5px; padding-right: 15px; padding-left: 15px; padding-bottom: 25px.

El valor del padding se puede indicar con una unidad absoluta (px, mm, ..) o con una unidad relativa (% , em, ...). También se puede heredar del elemento contenedor con el valor inherit.

Padding adyacentes. Al contrario que sucedía con los márgenes, con el padding no hay interacción con las cajas adyacentes ya que el padding está confinado dentro del borde de su caja, además no puede tomar valores negativos.

Fondo

Dentro del modelo de cajas el fondo está formado por dos elementos opcionales, la imagen de fondo y el color de fondo. El contenido está delante de ambos. Por defecto, tanto la imagen como el color de fondo llegan hasta el border, a continuación tenemos el margin que separa esta caja de las adyacentes. En el siguiente esquema hemos representado todos estos elementos.



El contenido, la imagen de fondo y el color de fondo no pueden sobrepasar el border, que puede ser de color o transparente.

Como ya hemos visto entre el contenido y el border puede existir el padding. El padding es transparente.

Entre el border y la siguiente caja puede existir un margin que es siempre transparente.

El contenido está siempre delante pero si tiene áreas transparentes, se puede ver lo que hay detrás. Por ejemplo el texto de las etiquetas es transparente entre los caracteres y permite ver la imagen de fondo. Está a su vez puede tener áreas transparentes que permitan ver el color de fondo.

Background-color

Este atributo opcional define el color de fondo, el valor debe ser un color escrito de una de las formas válidas que vimos anteriormente en el punto Colores. Este atributo se puede aplicar a la mayor parte de las etiquetas, div, p, table, tr, td, li, ... Por ejemplo, en un párrafo;

background-color: lightblue

Si utilizamos un color con una cierta transparencia, se mostrará el color de fondo de la página, o del elemento contenedor. Recuerda que se pueden mostrar transparencia, por ejemplo, con el formato RGBA.

Background-image

Este atributo opcional define la imagen de fondo, el valor debe ser una imagen, de la forma url(imagen). Este atributo se puede aplicar a bastantes etiquetas. Por ejemplo, en un párrafo;



background-image: url(graficos/mediana.png)

Como ya hemos dicho anteriormente, si la imagen es más pequeña que la caja, la imagen se repetirá hasta cubrir toda el área del contenido. De esta forma se pueden utilizar imágenes pequeñas para dar un aspecto de textura al fondo de la caja. Si la imagen tiene zonas transparentes, se verá el color de fondo.

Podemos especificar más de una imagen, separadas por comas, la primera que escribamos se verá delante y el resto se irán colocando detrás.

Podemos especificar atributos para controlar la repetición y posición de la imagen de fondo.

background-repeat

Este atributo indica la forma en que se repite la imagen de fondo. Puede tener los siguientes valores: repeat, no-repeat, repeat-x, repeat-y, space, round.

Valor repeat, es el valor por defecto, la imagen se repite en el eje x (horizontalmente) y en el eje y (verticalmente) hasta cubrir el área del contenido. Al repetir una imagen se pueden conseguir efectos de textura y otro tipo de efectos.

Valor no-repeat, la imagen no se repite, por defecto se alinearán a la izquierda, pero podemos controlar la alineación con el atributo background-position.

Valor repeat-x, la imagen se repite solamente en el eje x (horizontalmente) hasta llegar a la anchura del contenido.

Valor repeat-y, la imagen se repite solamente en el eje y (verticalmente) hasta llegar a la altura del contenido.

Valor space, la imagen se repite en ambos ejes hasta cubrir el área del contenido, pero no se recorta en ningún caso, en su lugar se rellena con espacios.

Valor round, la imagen se repite en ambos ejes hasta cubrir el área del contenido, pero no se recorta en ningún caso, en su lugar se modifica la forma para adaptarla al espacio disponible, es decir la imagen se estira o achata un poco para ajustarla sin tener que recortarla.

Background-position

Este atributo indica la posición de la imagen de fondo respecto a la esquina superior derecha. La posición se da con una pareja de coordenadas, es decir, un valor para la posición horizontal (con left, center, right) y otro valor para la posición vertical (top, bottom, center), si solo se da un valor,



el segundo se asume que es center. También pueden dar los valores de la posición con un valor numérico o con porcentaje sobre la anchura o altura.

Por ejemplo: background-position: right bottom; coloca la imagen en la esquina inferior derecha.

Por ejemplo: background-position: 30% 50px; coloca la imagen en el 30% del espacio horizontal y a 50px respecto de la parte superior.

Background-clip

La propiedad background-clip determina el área sobre la que se coloca la imagen de fondo. Como habíamos dicho, por defecto, el fondo llega hasta el border, sin sobrepasarlo, pero podemos cambiar este comportamiento con nuevas opciones. Esta propiedad puede tomar estos tres valores:

1- border-box. El fondo llega hasta el borde exterior del border, pero no lo sobrepasa. El border será visible si tiene anchura mayor de cero, el border se sitúa siempre por encima del fondo. Hemos dibujado un borde punteado para que se vea que el fondo llega hasta el límite exterior del border. Es la opción por defecto.

2- padding-box. El fondo llega hasta el interior del border, no cubre el border, pero sí cubre el padding

3- content-box. El fondo cubre sólo el área de contenido. Es decir llega el inicio del padding, sin cubrirlo.

Background-attachment

La propiedad background-attachment indica cuando la imagen de fondo queda fija o se mueve con el scroll de la página. Hacemos scroll cuando desplazamos la página con la barra lateral, o con el dedo en una pantalla táctil. Tiene estos valores scroll, fixed y local.

- scroll, la imagen de fondo se mueve junto con el resto de la página al hacer scroll. Es el valor por defecto.

- fixed, la imagen de fondo permanece fija al hacer scroll.

- local, si la imagen de fondo está dentro de un elemento que tiene su propio scroll, la imagen de fondo se moverá dentro de ese elemento al hacer scroll,

Background-size

Permite recalcular el tamaño de la imagen de fondo dinámicamente. Podemos hacerlo de dos formas, con los valores auto, cover y contain; o con los valores de ancho y alto en pixels.



1- auto, la imagen se muestra con su tamaño original. Es la opción por defecto.

2- cover, hace lo necesario para que la imagen ocupe todo el contenedor. Si la imagen es más pequeña, la redimensiona; si es más grande la recorta.

3- contain, la imagen se verá completa, con sus proporciones originales. Si es necesario, repetirá la imagen, vertical y horizontalmente.

4- ancho y alto, damos dos valores de longitud para fijar el ancho y el alto de la imagen. Por ejemplo, en este caso: background-size: 300px 225px

Forma compacta de background

Las propiedades del fondo que hemos visto se pueden indicar de una forma compacta, escribiendo la propiedad background seguida de los valores correspondientes (color, imagen, posición, repetición, ...) separados por un espacio en blanco.

En principio no importa el orden en que coloquemos los valores, siempre que no haya confusión. Por ejemplo, tanto con estos valores:

```
background: lightblue url(graficos/aulacli_cuadrada_mediana.png) no-repeat center padding-box
```

Como con estos valores, (los mismos, ordenados de forma diferente):

```
background: no-repeat url(graficos/aulacli_cuadrada_mediana.png) lightblue center padding-box;
```

Obtendríamos el mismo resultado.

Bordes

Los bordes de un elemento de HTML se definen con tres atributos básicos: estilo, anchura y color. Veamos las propiedades CSS para definirlos. Luego veremos que se pueden definir de forma compacta con el atributo border y tres valores.

border-style

El estilo del borde se refiere a la forma del borde y puede tomar estos valores: none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset.

Se puede establecer un estilo para cada uno de los cuatro bordes (superior, derecho, inferior, izquierdo) o un único valor para todos ellos. Hay varias formas de hacerlo:

- un valor: border-style: solid, se aplica el estilo a los cuatro bordes.



- dos valores: border-style: solid dotted, se aplica el primer valor al borde superior e inferior, y el segundo valor al borde derecho e izquierdo.

- tres valores: border-style: solid dotted double, se aplica el primer valor al borde superior, el segundo valor al borde derecho e izquierdo, y el tercer valor al borde inferior.

- cuatro valores: border-style: solid dotted double dashed, se aplica el primer valor al borde superior, el segundo valor al borde derecho, el tercer valor al borde inferior, y el cuarto valor al borde izquierdo,

También se pueden aplicar de forma explícita valores a cada borde con las propiedades: border-top-style, border-right-style, border-bottom-style, border-left-style.

Border-width

El ancho del borde puede tomar valores numéricos absolutos y relativos (px, mm, em,...) y los valores predeterminados: thin, medium, thick.

Se puede establecer un ancho para cada uno de los cuatro bordes (superior, derecho, inferior, izquierdo) o un único valor para todos ellos. Hay varias formas de hacerlo:

- un valor: border-width: 15px, se aplica el estilo a los cuatro bordes.

- dos valores: border-width: 10px 0.9em, se aplica el primer valor al borde superior e inferior, y el segundo valor al borde derecho e izquierdo.

- tres valores: border-width: 10px 0.9em 5mm, se aplica el primer valor al borde superior, el segundo valor al borde derecho e izquierdo, y el tercer valor al borde inferior.

- cuatro valores: border-width: 10px thin medium thick, se aplica el primer valor al borde superior, el segundo valor al borde derecho, el tercer valor al borde inferior, y el cuarto valor al borde izquierdo,

También se pueden aplicar de forma explícita valores a cada borde con las propiedades: border-top-width, border-right-width, border-bottom-width, border-left-width.

Border-color

El color del borde se puede dar con los distintos tipos de formatos para colores que ya vimos (RGB, nombres_colores, HSL, ...).



Se puede establecer un color para cada uno de los cuatro bordes (superior, derecho, inferior, izquierdo) o un único valor para todos ellos. Hay varias formas de hacerlo que son las que ya hemos indicado para la anchura y el estilo del borde.

- un valor: `border-color: royalblue`, se aplica el estilo a los cuatro bordes.
- dos valores: `border-color: royalblue #6434ea`, se aplica el primer valor al borde superior e inferior, y el segundo valor al borde derecho e izquierdo.
- tres valores: `border-color: royalblue rgb(0, 100, 0) #6434ea`, se aplica el primer valor al borde superior, el segundo valor al borde derecho e izquierdo, y el tercer valor al borde inferior.
- cuatro valores: `border-color: royalblue rgb(0, 100, 0) hsl(187, 82%, 30%) #6434ea`, se aplica el primer valor al borde superior, el segundo valor al borde derecho, el tercer valor al borde inferior, y el cuarto valor al borde izquierdo,

También se pueden aplicar de forma explícita valores a cada borde con las propiedades: `border-top-color`, `border-right-color`, `border-bottom-color`, `border-left-color`.

Border-radius

Con esta propiedad las esquinas de los bordes se pueden hacer redondeadas. El valor está compuesto por dos valores el radio horizontal y el radio vertical. Si sólo se da un valor la esquina tendrá forma redondeada, y si se dan dos valores forma de elipse.

Ejemplo con un valor de radio, `border-radius: 15px`.

Ejemplo con dos valores de radio, `border-radius: 15px / 30px`:

Se puede establecer un radio para cada uno de las cuatro esquinas (superior, derecho, inferior, izquierdo) o un único valor para todas ellas. Hay varias formas de hacerlo que son las que ya hemos indicado para la anchura y el estilo del borde. Con un valor, dos valores, tres valores y cuatro valores.

También se pueden aplicar de forma explícita valores a cada borde con las propiedades: `border-top-left-radius`, `border-top-right-radius`, `border-bottom-left-radius`, `border-bottom-right-radius`.

Cuando se utiliza la forma explícita los dos valores para cada esquina indican el radio horizontal y vertical, no hace falta poner una barra.

Por ejemplo: `border-top-left-radius: 15px 25px; border-top-right-radius: 15px 25px;`

Cuando se utiliza la forma compacta, una barra separa el radio horizontal y vertical.



Por ejemplo: border-radius: 15px 15px / 25px 25px

Los valores también se pueden dar como porcentajes de la longitud de la caja contenedora. En ese caso la forma de la esquina variará según modificamos el ancho de la ventana. Con estos valores: border-radius: 50% 50%, el borde irá desde una elipse a una esfera:

Imágenes con bordes redondeados. Podemos aplicar los bordes redondeados a las imágenes, incluso con un porcentaje para que el redondeo sea proporcional al tamaño de la imagen. Por ejemplo: ``

Forma compacta del borde

Como sucedía para el fondo, existe una forma mucho más compacta de definir el borde, escribiendo la propiedad border seguida de los valores correspondientes (anchura, estilo y color) separados por un espacio en blanco.

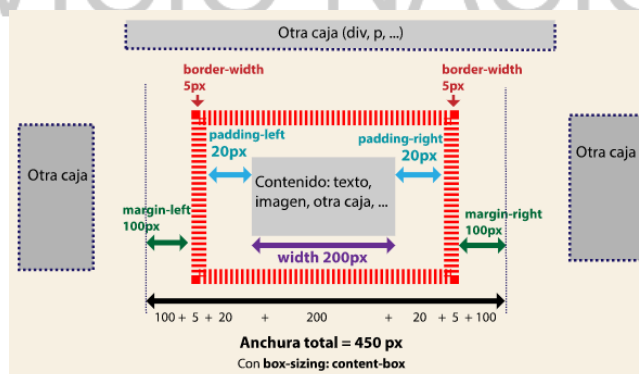
Por ejemplo: border: 5px solid royalblue ;

Cálculo de la anchura y altura

Como ya hemos visto los atributos width y height definen la anchura y altura de los elementos HTML, se utilizan mucho para definir el diseño de la página, sobre todo en los elementos contenedores, (div, p, etc.) Es importante tener en cuenta a qué se refieren estos valores, ya que en ocasiones se comete el error de creer que incluyen partes que no incluyen en realidad.

Los atributos width y height definen la anchura y altura de los elemento HTML, pero sólo incluyen el contenido de la etiqueta, no incluyen el padding, ni el margin, ni el border.

Vamos a explicar el caso de width con el siguiente esquema (el height sería similar):





El esquema corresponde a la definición de un contenedor (por ejemplo div) con estos atributos:

```
div { width: 200px; border-width: 5px; padding: 20px; margin: 100px }
```

Por defecto (sin definir el box-sizing) deberemos reservar un espacio en nuestro diseño de la página de 450 px., ya que al width del contenido hay que sumar el padding y el border, y luego añadir el margin. Es decir, 200 px, que ocupa el contenido, más 100 px, por cada lado del margin; otros 5 px. por cada lado del border; otros 20 px. por cada lado del padding.

Esta forma de calcular el ancho del modelo de caja no es la más intuitiva, y ha provocado numerosos errores en el diseño, en unos casos por su complejidad y en otros casos por desconocimiento sobre cómo se calcula la anchura.

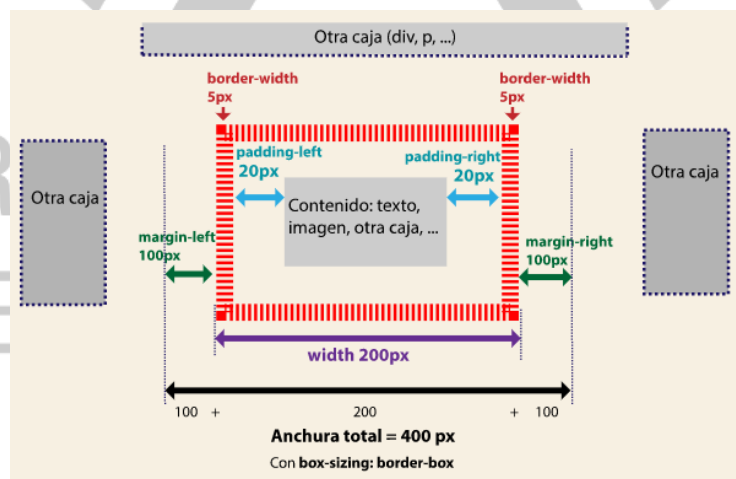
Afortunadamente existe la propiedad box-sizing que vamos a ver a continuación.

Propiedad box-sizing

La propiedad box-sizing indica cómo se calcula el ancho y alto de la etiqueta correspondiente. Tiene dos valores: content-box y border-box:

- content-box, el ancho y el alto se calculan teniendo en cuenta sólo el contenido, sin incluir el border y el padding. Es el valor por defecto.
- border-box, el ancho y el alto se calculan teniendo en cuenta el contenido y también el border y el padding. Es decir, el border y el padding están incluidos en el ancho y alto.

El caso de content-box lo hemos explicado en el punto anterior, ahora vamos a ver un esquema para ver el caso de border-box:



El esquema corresponde a la definición de un contenedor (por ejemplo div) con estos atributos:



div { box-size: border-box; width: 200px; border-width: 5px; padding: 20px; margin: 100px }

El espacio que debemos reservar en nuestro diseño de la página es de 400 px., que corresponden al width del contenido y al margin. Es decir, 200 px, que ocupa el contenido, más 100 px, por cada lado del margin. En este caso los 5 px. por cada lado del border, y los 20 px. por cada lado del padding, ya están incluidos en el width del contenido.

A continuación puedes ver un primer ejemplo con box-sizing: border-box y un segundo ejemplo con el valor por defecto, box-sizing: content-box.

El resto de valores son igual para ambos ejemplos: border-width: 5px; border-style: solid; border-color: royalblue; padding: 20px; width: 200px; margin-left: 100px; margin-right: 100px;

Desbordamiento. overflow

La propiedad overflow indica cómo se muestra el contenido cuando es mayor que las dimensiones de la caja que lo contiene. Puede tomar cuatro valores: visible, hidden, scroll, auto.

1- visible, el contenido desborda la caja y se superpone con la caja siguiente. Es el valor por defecto.

2- hidden, el contenido que desborda de la caja se oculta, no se muestra.

3- scroll, el contenido que desborda la caja no se muestra pero se habilitan las barras de scroll para poder verlo (en los dispositivos táctiles se puede hacer scroll con el dedo).

4- auto, si el contenido desborda la caja no se muestra pero se habilitan las barras de scroll que sean necesarias para poder verlo (en los dispositivos táctiles se puede hacer scroll con el dedo).

Pueden consultar mas a través del siguiente [link](#)

Taller

Después de leer el material compartido deberán realizar una página web que tenga la siguiente estructura y en el cual apliquen todo lo mencionado en el presente material:

Tengan en cuenta que la elección de colores es libre, se deben manejar diferentes tipos de bordes y aplicar en los casos que corresponda color a los botes y la forma de los mismos. Deben manejar transparencias y opacidad y todos aquellos que uds consideren necesarios para que sea una diseño llamativo



imagen - logo

Información de la empresa utilizando diferentes tamaños y colores de texto

Productos y /o Servicios ofertados por la empresa

Imágenes y lista de productos y/o Servicios ofertados por la empresa en tres columnas

Mucho éxitos!!!

SERVICIO NACIONAL
DE APRENDIZAJE