

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Curso:

PROGRAMADOR WEB INICIAL

Módulo 1:

DISEÑO WEB

Unidad 4:

Maquetación Web: modelo de cajas

Presentación

Tal como vimos en la unidad anterior, podemos considerar a todos los elementos HTML como cajas.

Ese concepto de que “todo es una caja” da lugar a algo llamado en web como *box model* o *modelo de cajas*. Sin importar si son de línea o de bloque, todas las etiquetas tienen algunas propiedades en común: el contenido, el relleno, el borde, y el margen.

En esta unidad conoceremos qué es el *modelo de cajas* en CSS, veremos su funcionamiento y terminología.

Además, crearemos nuestros primeros diseños, haciendo uso de sus propiedades.

Objetivos

Que los participantes logren...

- Comprender las cajas y sus propiedades.
- Modelar la página web.
- Conocer las posiciones de un elemento.

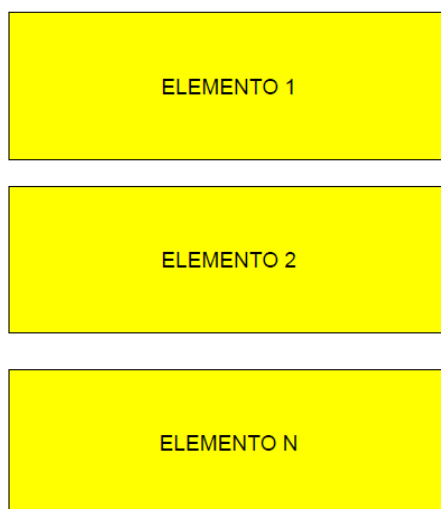
Bloques temáticos

1. Modelo de cajas
2. Tipos de elementos de cajas
3. Partes de cada caja
 - a. Anchura
 - b. Altura
 - c. Margen
 - d. Relleno
 - e. Bordes
 - i. Anchura
 - ii. Color
 - iii. Estilo
4. Margen relleno, bordes y modelo de cajas
5. Fondos
6. Posicionamiento
 - a. Propiedad OVERFLOW
 - b. Propiedad Z-INDEX

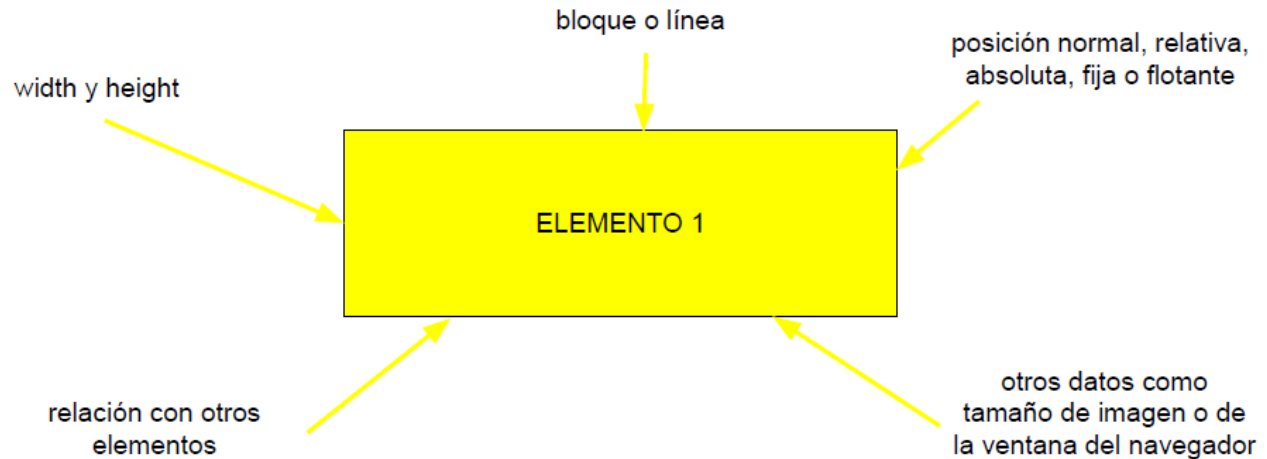
MODELO DE CAJAS

El **modelo de cajas** o “**box model**” es el comportamiento de **hojas de estilos CSS** que coloca todos los elementos de las páginas web en “cajas rectangulares”

Las cajas se crean de forma automática y cada vez que se agrega una etiqueta HTML, se crea una nueva caja rectangular que posee todos los contenidos de ese elemento.



Factores que tienen en cuenta al crear cada caja:

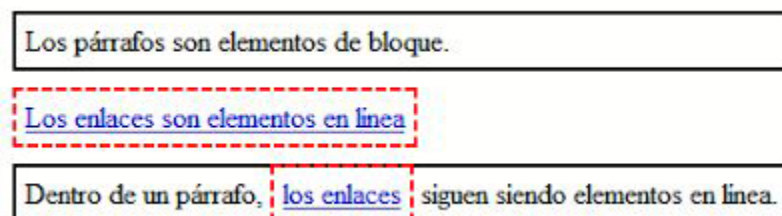


Recordamos los tipos de elementos...

- **Bloque:** comienzan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea. Los elementos bloque no pueden insertarse en elementos de línea. Sólo dentro de otros elementos bloque.
- **Línea:** pueden no empezar en una línea y ocupan el espacio necesario para mostrar sus contenidos. Un elemento de línea puede aparecer dentro de un elemento bloque o línea.

Elementos y cajas

- **Párrafo:** es un elemento de tipo bloque.
- **Enlace:** es un elemento de tipo línea. Puede tener otros elementos en la misma línea.



La siguiente imagen muestra las tres cajas rectangulares que crean las tres etiquetas HTML que incluye la página:

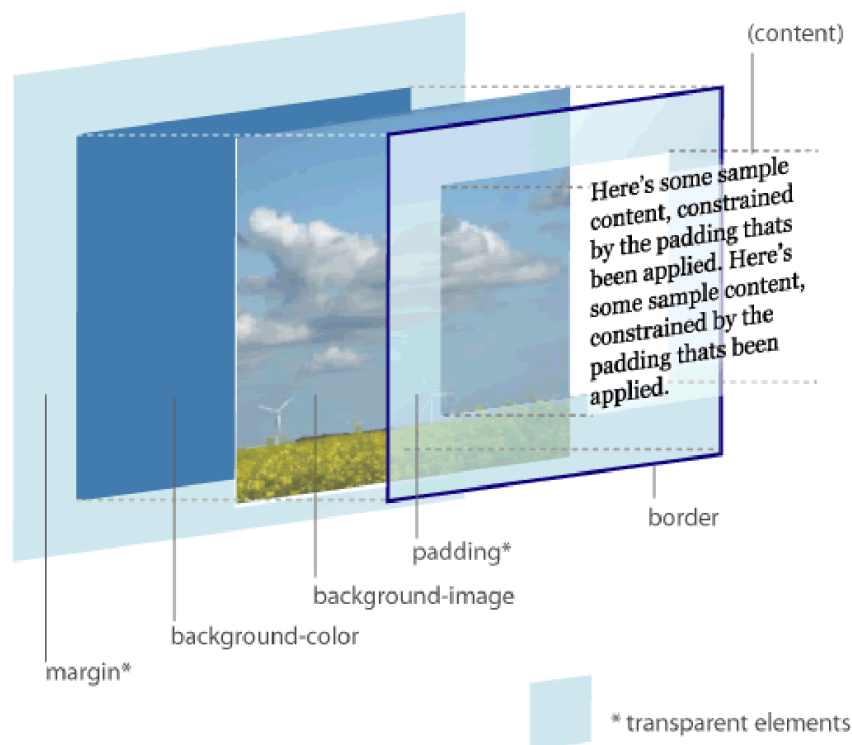
<code><p>Párrafo de texto con algunas palabras resaltadas</p></code>	Párrafo de texto con algunas palabras resaltadas
<code><p>Otro párrafo</p></code>	Otro párrafo

Partes de cada caja

Recordemos que los navegadores crean y colocan las cajas de forma automática, pero podremos modificar sus características manualmente con diferentes propiedades de CSS.

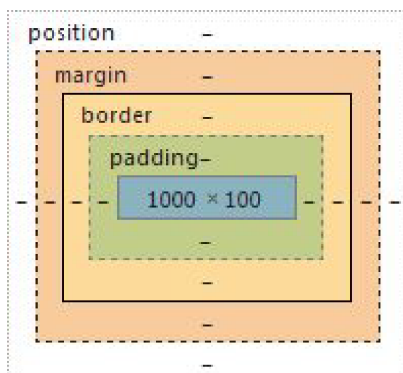
En la siguiente imagen podremos ver las diferentes partes que componen las cajas:

THE CSS BOX MODEL HIERARCHY



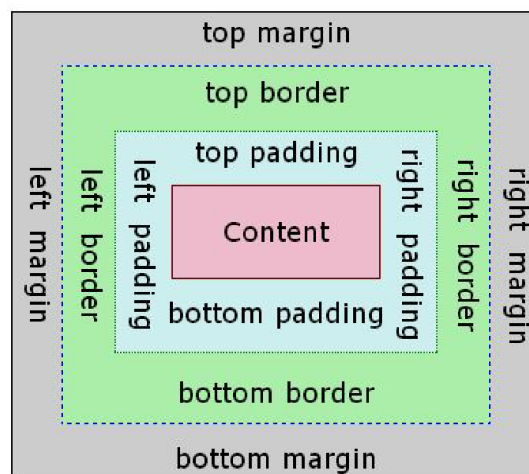
(Esquema utilizado con permiso de <http://www.hicksdesign.co.uk/boxmodel/>)

Desarrollaremos cada una de las partes a continuación.



- **Contenido (content):** es el contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
- **Relleno (padding):** es el espacio libre que existe entre el contenido y el borde. Este es opcional y puede estar o no.
- **Borde (border):** es la línea que encierra completamente al contenido y su relleno.
- **Imagen de fondo (background image):** es la imagen de fondo que se muestra por detrás del contenido y el espacio de relleno.
- **Color de fondo (background color):** es el color de fondo que se muestra por detrás del contenido y el espacio de relleno, este también se visualizará en el caso que la imagen no se cargue correctamente..
- **Margen (margin):** es la separación que existe entre la caja actual y el resto de cajas adyacentes. Es opcional al igual que el relleno.

Debemos tener en cuenta que el **relleno** y el **margen** son transparentes, entonces se visualizará el color o imagen de fondo elegido. Si definimos ambos, siempre la imagen tendrá prioridad sobre el color. En el caso que no se descargue correctamente la imagen de fondo por ejemplo, como mencionamos anteriormente, se visualizará el color directamente. Otra manera de poder ver el color de fondo es utilizando imágenes con transparencias, y así lograr diferentes efectos.



Anchura - width.

La propiedad **width** es la que define la anchura de las cajas. No admite valores negativos y si colocamos los valores en porcentaje se van a calcular con respecto a la **anchura** de su elemento padre, es decir, del elemento que lo contiene.

- El valor **inherit** indica que la anchura del elemento se hereda de su elemento padre.
- El valor **auto** indica que el navegador debe calcular automáticamente la anchura del elemento, teniendo en cuenta sus contenidos y el sitio disponible en la página. Es el valor por defecto si no se coloca un valor específico.

Altura - height

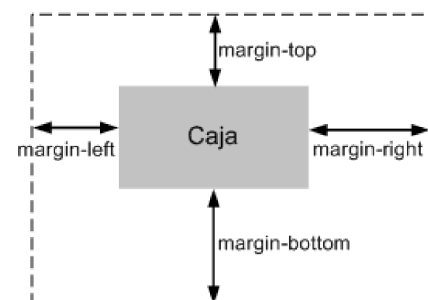
Funciona de la misma forma que la anchura, se la denomina **height**. No admite valores negativos y si se indica un **porcentaje**, se toma como referencia la altura del elemento padre. Posee los mismos valores que **width**,

- **inherit**
- **auto**

Margen - margin

Con esta propiedad definiremos los márgenes verticales y horizontales de un elemento. Tenemos una para cada lado, y establecen la **separación entre el borde lateral de la caja y el resto de cajas adyacentes**:

- **margin-top**
- **margin-right**
- **margin-bottom**
- **margin-left**

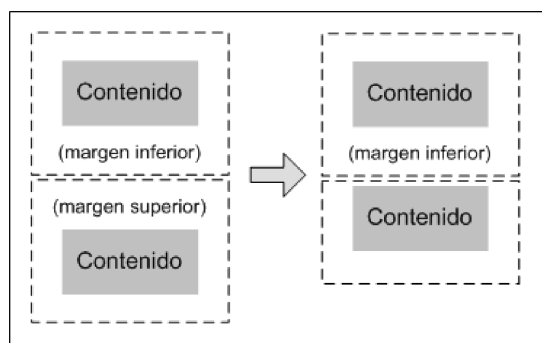


Los **márgenes verticales** (margin-top y margin-bottom) sólo podrán aplicarse a los elementos de bloque y a las imágenes. Por otro lado, los **márgenes laterales** (margin-left y margin-right) se pueden aplicar a cualquier elemento.

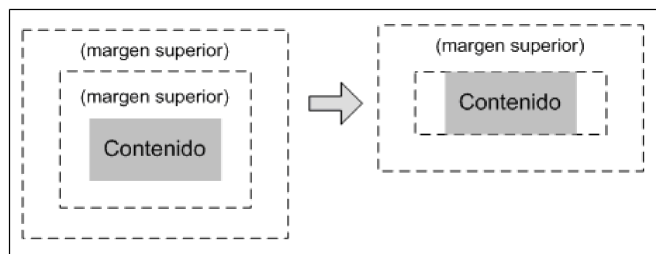
La propiedad que permite definir los cuatro márgenes de forma simultánea se denomina **margin** y admite entre uno y cuatro valores:

- **un valor**, definirá todos los márgenes con el mismo valor.
- **dos valores**, definirá con el primer valor, los márgenes superior e inferior, mientras que el segundo valor, define los márgenes laterales.
- **tres valores**, define con el primer valor el margen superior, el segundo valor será el correspondiente a los márgenes laterales y el tercer valor corresponde al margen inferior.
- **cuatro valores**, definirá los valores de los márgenes en el orden de las agujas del reloj. El primer valor será el margen superior, luego el margen derecho, seguido por el margen inferior y por último el margen izquierdo.

Tendremos diferentes comportamientos de los márgenes verticales y laterales.



Por ejemplo, cuando se juntan dos o más márgenes verticales, **se fusionan de forma automática** y la altura del nuevo margen será igual a la altura del margen más alto de los que se han fusionado.

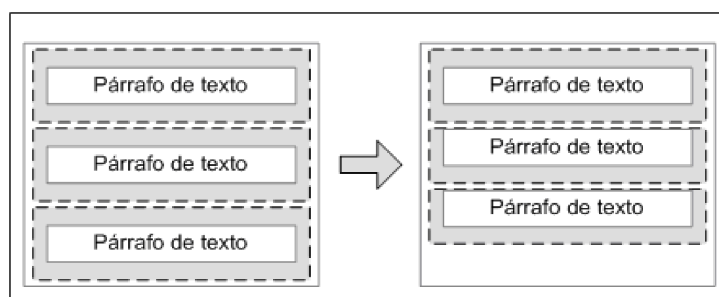


De la misma forma, si un elemento está contenido dentro de otro elemento, **sus márgenes verticales se fusionan** y resultan en un nuevo

margen de la misma altura que el mayor margen de los que se han fusionado.

Podremos evitar este comportamiento **añadiendo un pequeño relleno** (padding: 1px) **o un borde** (border: 1px solid transparent) al elemento contenedor.

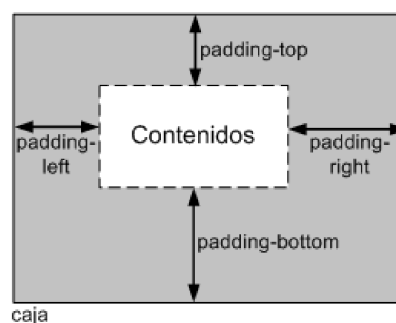
Esto se realiza de manera automática para dar uniformidad a nuestra página. Por ejemplo, si tenemos varios párrafos y no se diera este comportamiento; el primer párrafo no sería homogéneo respecto al resto.



Relleno - padding

Cómo vimos en los márgenes, también podremos definir los espacios de relleno de cada lado de las cajas. Establecen **la separación entre el lateral de los contenidos y el borde lateral de la caja**:

- padding-top
- padding-right
- padding-bottom
- padding-left



La propiedad que permite definir de forma simultánea los cuatro rellenos se denomina **padding**.

La propiedad **padding** admite entre uno y cuatro valores, con el mismo significado que el de la propiedad **margin**.

Bordes - border

Cómo hemos visto anteriormente, podremos definir en CSS el estilo de los bordes de cada elemento. Se podrá modificar su anchura, su color y su estilo.

Anchura

Poseemos una propiedad para cada lado de la caja:

- **border-top-width**
- **border-right-width**
- **border-bottom-width**
- **border-left-width**

La anchura de los bordes se podrá indicar con cualquier unidad de medida definida o utilizando las siguientes palabras clave:

- **thin** (borde delgado),
- **medium** (borde normal),
- **thick** (borde ancho).

Color

Podremos definir también el color para cada lado:

- **border-top-color**
- **border-right-color**
- **border-bottom-color**
- **border-left-color**

Estilo

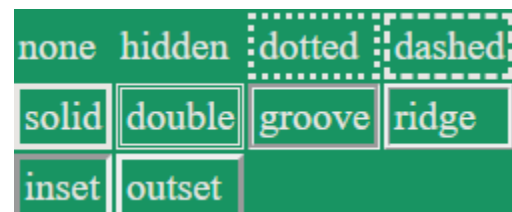
Por último, CSS permite establecer el estilo de cada uno de los bordes mediante las siguientes propiedades:

- **border-top-style**

- **border-right-style**
- **border-bottom-style**
- **border-left-style**

El estilo de los bordes sólo se puede indicar utilizando las palabras reservadas. Entre los más usados están los siguientes:

- **none**: valor por defecto, sin borde.
- **solid**
- **dashed**
- **double**
- **dotted**



La propiedad borde admite los 3 valores en la misma regla:

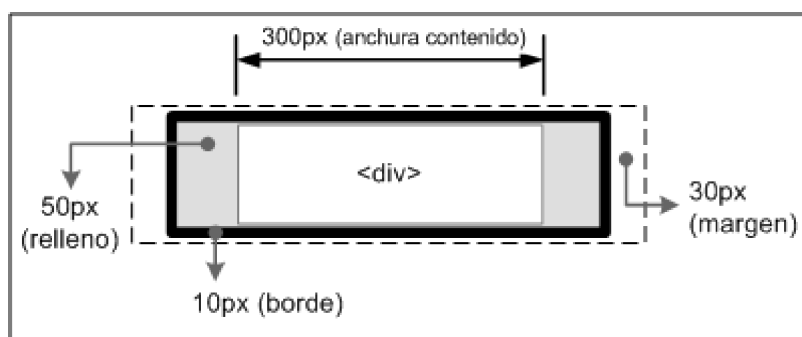
```
td { border: 5px dotted #660000; }
```

Margen, relleno, bordes y modelo de cajas

La **anchura** y **altura** de un elemento no solamente se calculan teniendo en cuenta sus propiedades width y height. El margen, el relleno y los bordes establecidos a un elemento determinan la anchura y altura final del elemento. En el siguiente ejemplo se muestran los estilos CSS de un elemento:

```
li {
    width: 300px;
    padding-left: 50px;
    padding-right: 50px;
    margin-left: 30px;
    margin-right: 30px;
    border: 10px solid black;
}
```

La **anchura total** con la que se muestra el elemento no son los 300 píxel indicados en la propiedad width, sino que se tienen en cuenta todos sus márgenes, rellenos y bordes:



De esta forma, la anchura del elemento en pantalla sería igual a la suma de la anchura original, los márgenes, los bordes y los rellenos:

$$30px + 10px + 50px + 300px + 50px + 10px + 30px = 480 \text{ píxel}$$

Así, la anchura/altura establecida con CSS siempre hace referencia a la anchura/altura del contenido. **La anchura/altura total del elemento debe tener en cuenta además los valores del resto de partes que componen la caja del box model.**

Fondos

Podemos también definir el fondo de la caja del elemento. Este puede ser un color simple o una imagen. Recordemos que el fondo solamente se visualiza en el área del contenido y su relleno, ya que los márgenes siempre son transparentes.

Si queremos colocar un color o imagen de fondo a la página entera debemos establecer un fondo al elemento **`<body>`**. Como todos los elementos inicialmente tienen un fondo transparente, todos nuestros elementos se visualizarán con el mismo fondo. Aprenderemos cómo especificar el fondo a los elementos para

personalizarlos a gusto.

- **background-color,**
- **background-image:** Establece una imagen como fondo para los elementos
Valores: <url> | none | inherit
- **background-repeat,**
- **background-attachment,**
- **background-position.**
- **background-image:**

Las imágenes de fondo se indican a través de URL.. Se recomienda crear una carpeta de imágenes dentro del mismo directorio que los archivos CSS. Así, las imágenes correspondientes al diseño se mantienen separadas del resto de imágenes del sitio y el código CSS es más sencillo y más fácil de mantener.

Si la imagen que se quiere mostrar es demasiado grande, se mostrará la parte de imagen que coincide con el tamaño del elemento.

Si la imagen es más pequeña que el elemento, CSS la repite horizontal y verticalmente hasta llenar el fondo del elemento. Este comportamiento es útil para establecer un fondo complejo a una página web entera.

El siguiente ejemplo utiliza una imagen muy pequeña para establecer un fondo complejo a toda una página:

Imagen original



Reglas CSS

```
body {  
    background-image:url(imagenes/fondo.gif);  
}
```

Con una imagen muy pequeña se consigue cubrir completamente el fondo de la página, lo que nos hará ahorrar ancho de banda.

En ocasiones, queremos que la imagen de fondo no se repita horizontal y verticalmente. En ese caso utilizaremos la propiedad **background-repeat** que nos permite determinar la manera en la que se repetirán las imágenes de fondo.

- El valor **repeat** indica que la imagen se debe repetir en todas direcciones y por tanto, es el comportamiento por defecto.
- El valor **no-repeat** muestra una sola vez la imagen y no se repite en ninguna dirección.
- El valor **repeat-x** repite la imagen sólo horizontalmente y
- el valor **repeat-y** repite la imagen solamente de forma vertical.

```
body {  
    background-image:url(imagenes/fondo.gif);  
    background-repeat: repeat-x;  
}
```

Posicionamiento

CSS define 5 modelos diferentes para posicionar una caja:

- **Normal o Estático (static):** el que utilizan los navegadores si no se indica lo contrario.
- **Flotante:** ubica un elemento al lado izquierdo o derecho de su contenedor, permitiendo a los elementos de texto y en línea aparecer a su costado. El elemento es removido del normal flujo de la página, aunque aún sigue siendo parte del flujo.

Los elementos **en línea** hacen sitio a las cajas flotantes adaptando su anchura al espacio libre dejado por la caja desplazada.

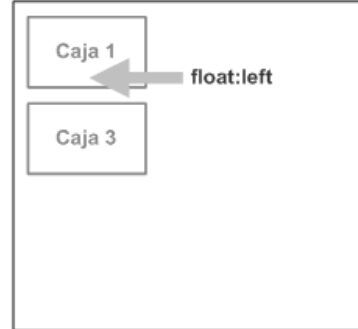
Los elementos **de bloque** no les hacen sitio, pero sí que adaptan sus contenidos para que no se solapen con las cajas flotantes.

Elemento contenedor



Posicionamiento normal

Elemento contenedor



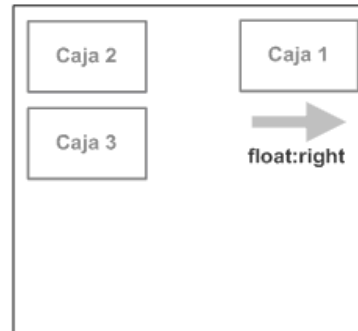
Posicionamiento float de la caja 1

Elemento contenedor



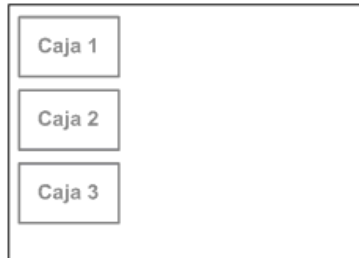
Posicionamiento normal

Elemento contenedor



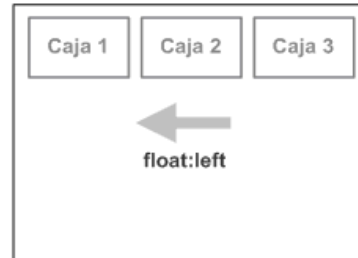
Posicionamiento float de la caja 1

Elemento contenedor

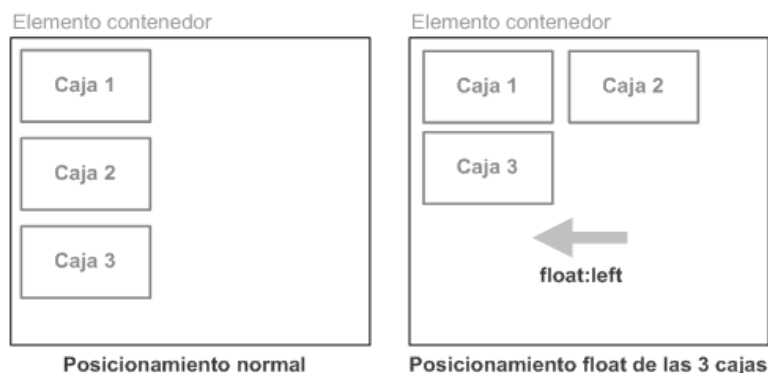


Posicionamiento normal

Elemento contenedor



Posicionamiento float de las 3 cajas



- **Fijo:** convierte una caja en un elemento inamovible. Su posición en la pantalla es siempre la misma sin depender del resto de elementos, o si el usuario sube o baja la página en la ventana del navegador.
- **Absoluto (absolute):** la posición de una caja es absoluta respecto de su elemento contenedor. Los demás elementos se ven afectados. Se pueden producir solapamientos.



- **Relativo (relative):** tomando como referencia el posicionamiento estático, la desplaza. Su desplazamiento no afecta al resto de las cajas. Se utiliza en conjunto con cuatro propiedades más: **bottom**, **top**, **right** y **left**.

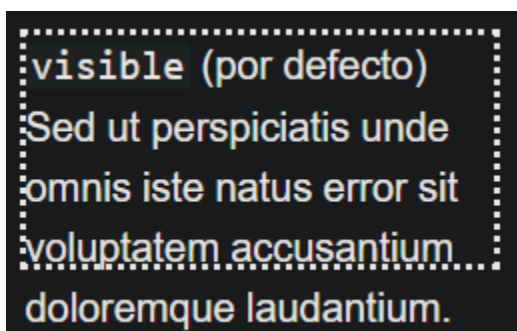


PROPIEDAD OVERFLOW

Los contenidos de un elemento se muestran en el espacio designado para ese elemento. Sin embargo, algunas veces el contenido no entra en el espacio reservado para ese elemento y se desborda. Normalmente esto ocurre cuando se estableció la anchura y altura del elemento contenedor mediante la propiedad width y/o height.

Es por esto que CSS nos brinda la propiedad **overflow**, la cual posee los siguientes parámetros:

- **visible:** Es el valor por defecto. El contenido no es recortado, podría ser dibujado fuera de la caja contenedora.



- **hidden:** El contenido es recortado y el sobrante se oculta. Sólo se visualiza la parte del contenido que entra dentro de la zona reservada.



```
overflow: hidden  
Sed ut perspiciatis unde  
omnis iste natus error sit  
voluptatem accusantium...
```

- **scroll:** El contenido es recortado y el navegador web usa las barras de desplazamiento, se haya recortado contenido o no.

```
overflow: scroll  
Sed ut perspiciatis unde  
omnis iste natus error sit
```

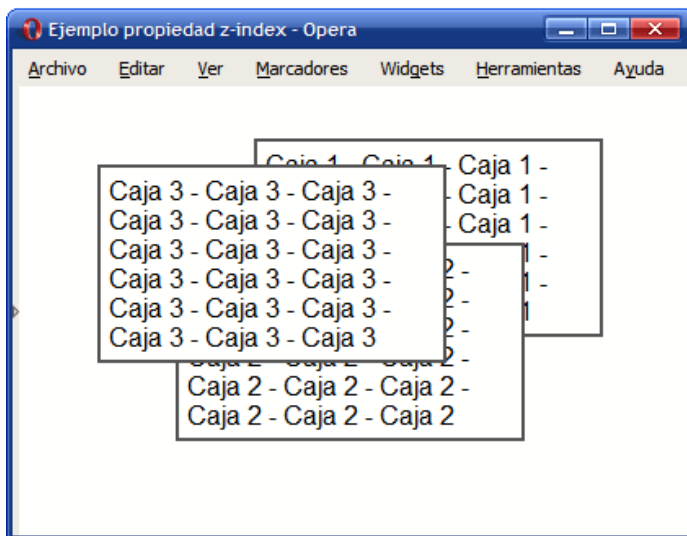
- **auto:** el comportamiento depende del navegador, aunque normalmente es el mismo que la propiedad scroll.

```
overflow: auto  
Sed ut perspiciatis unde  
omnis iste natus error sit  
voluptatem accusantium...
```

PROPIEDAD Z-INDEX

Podemos también controlar la posición de las cajas, como si fueran capas, indicando cuan "arriba" o "abajo" se encuentra la caja. De esta forma, es posible indicar como se muestran: delante o detrás de otras cajas cuando se producen solapamientos. La posición tridimensional de un elemento se establece sobre un **tercer eje llamado Z** y se controla mediante la propiedad **z-index**.

Debemos tener en cuenta, que sólo tiene efecto en los elementos posicionados, por lo que debemos colocar la propiedad junto con **position**. En el caso que no queramos modificar la posición original de la caja, debemos establecer la position como relative.



El valor que admite es un número entero. Se considera el número 0 (cero) como el nivel más bajo y cuanto más alto sea el valor numérico, más cerca del usuario se muestra la caja.

Por ejemplo: Un elemento con z-index: 10 se muestra por encima de los de z-index: 8 o z-index: 9, pero por debajo de elementos con z-index: 20 o z-index: 50.

En resumen

En esta unidad hemos abordado el modelo de cajas en CSS, conocimos cómo está conformada cada caja y como aplicar sus propiedades en nuestros diseños.

También vimos los distintos tipos de posicionamiento y conocimos las propiedades **OVERFLOW** que maneja todo el contenido que sobresalga del *border* (visible o no) y **Z-INDEX** para controlar la posición de las cajas.

Las características vistas hasta el momento nos permitirán ir definiendo nuestro modelo de cajas e implementarlo en nuestro sitio web.

Bibliografía utilizada y sugerida

- CSS. Propiedades de los textos. Recuperado de:
<https://www.eniun.com/atributos-propiedades-textos-css/>
- MDN Web Docs. Mozilla. Introducción al Modelo de Caja de CSS. Recuperado de:
https://developer.mozilla.org/es/docs/Web/CSS/CSS_Box_Model/Introduction_to_the_CSS_box_model
- The W3C CSS Validation Service. Recuperado de:
https://jigsaw.w3.org/css-validator/#validate_by_upload