

POSICIONAMIENTO HTML/CSS

Con HTML y sus etiquetas podemos marcar y estructurar los elementos de manera individual en las páginas webs: tablas, listas, enlaces, párrafos, imágenes, etc. Aunque combinando estas etiquetas podemos crear cualquier página web, no es posible hacer páginas con estructuras complejas.

Para crear páginas Web con columnas de contenidos y otro tipo de divisiones es necesario utilizar **HTML y CSS de manera combinada**.

1 POSICIONAMIENTO EN HTML

En concreto, el código HTML se encarga de agrupar los elementos de la página en diferentes divisiones en función de su finalidad: la zona de la cabecera de la página, la zona de contenidos, una zona lateral para el menú y otras secciones menores, la zona del pie de página, etc. Para agrupar y dar semántica al contenido HTML se utilizan las etiquetas `<head>`, `<section>`, `<article>` ...

Para posicionar los elementos, el HTML utiliza dos etiquetas que no hacen nada especial salvo actuar de contenedores: `` y `<div>`.

- Elementos de Posicionamiento en HTML

- `` se usa para no romper el flujo, es decir en principio todo va en la misma línea
- `<div>` sí rompe el flujo, por lo que va a una línea distinta

El elemento `span` es un contenedor genérico sin un significado semántico en particular. Es comúnmente utilizado para especificar un atributo `style` o `class` de manera aislada agrupar elementos en línea. Para bloque más grandes de contenido se usa el elemento `div` en su lugar.

El elemento `div` tiene su origen en la palabra *división*, ya que define zonas o divisiones en bloque dentro de una página HTML.

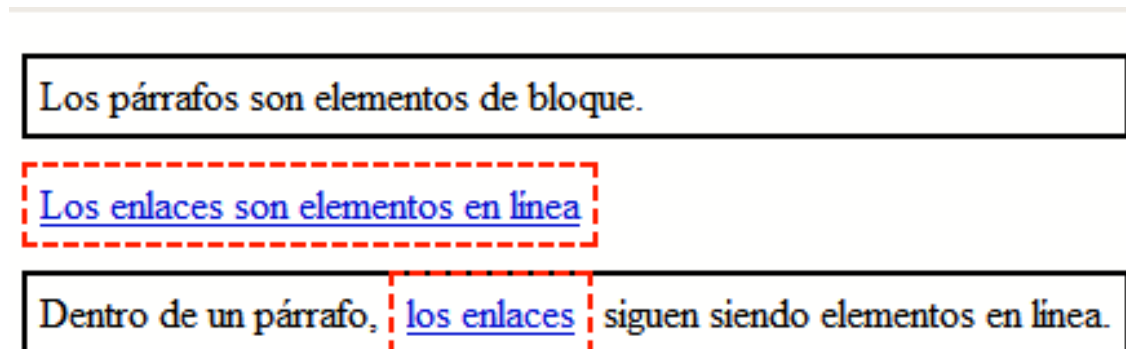
TIPOS DE ELEMENTOS HTML

El estándar HTML clasifica a todos sus elementos en dos grandes grupos: **elementos en línea** y **elementos de bloque**.

Los elementos de bloque (*"block elements"* en inglés) siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea. Por su parte,

los elementos en línea (*"inline elements"* en inglés) no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.

Debido a este comportamiento, el tipo de un elemento influye de forma decisiva en el espacio (caja en CSS) que el navegador crea para mostrarlo. Ejemplo:



Por sus características, los elementos de bloque no pueden insertarse dentro de elementos en línea y tan sólo pueden aparecer dentro de otros elementos de bloque. En cambio, un elemento en línea puede aparecer tanto dentro de un elemento de bloque como dentro de otro elemento en línea.

Los elementos en línea definidos por HTML son:

a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

Los elementos de bloque definidos por HTML son:

address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, i, index, menu, noframes, noscript, ol, p, pre, table, ul.

Posicionamiento de los elementos HTML

El posicionamiento normal de los elementos en la página (por defecto) es colocarse según el flujo normal del HTML, es decir según estén escritos en el propio código HTML.

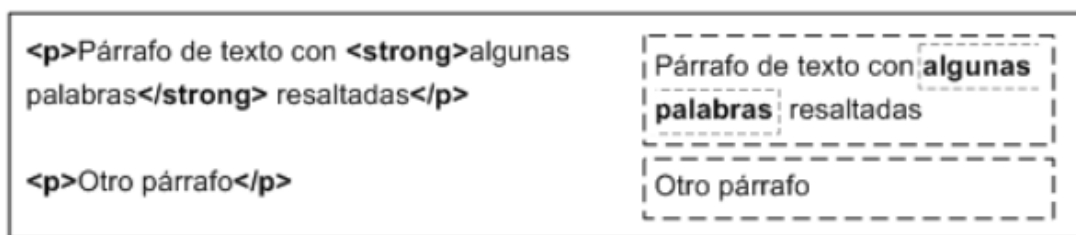
Para cambiar el posicionamiento ya es necesario incluir atributos CSS

2 POSICIONAMIENTO EN CSS

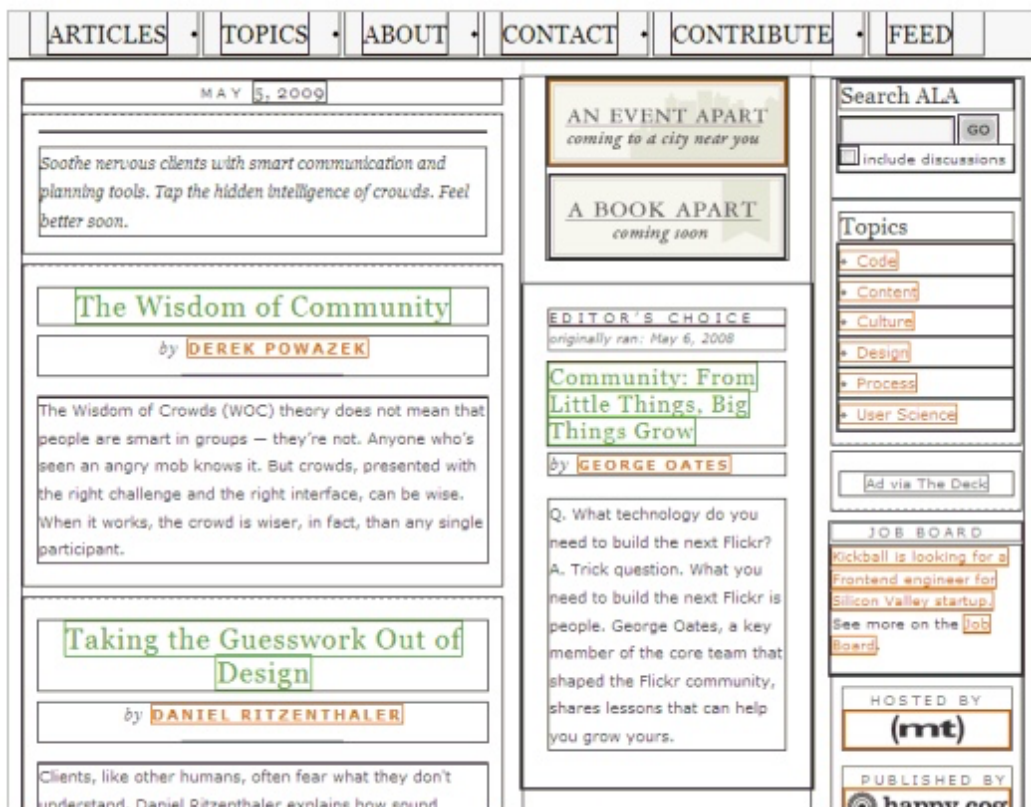
2.1 MODELO DE CAJAS

El modelo de cajas o *"box model"* es la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web. El modelo de cajas es el comportamiento de CSS que hace que todos los elementos de las páginas se representen mediante **cajas rectangulares**.

Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento.



Las cajas de las páginas no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde.

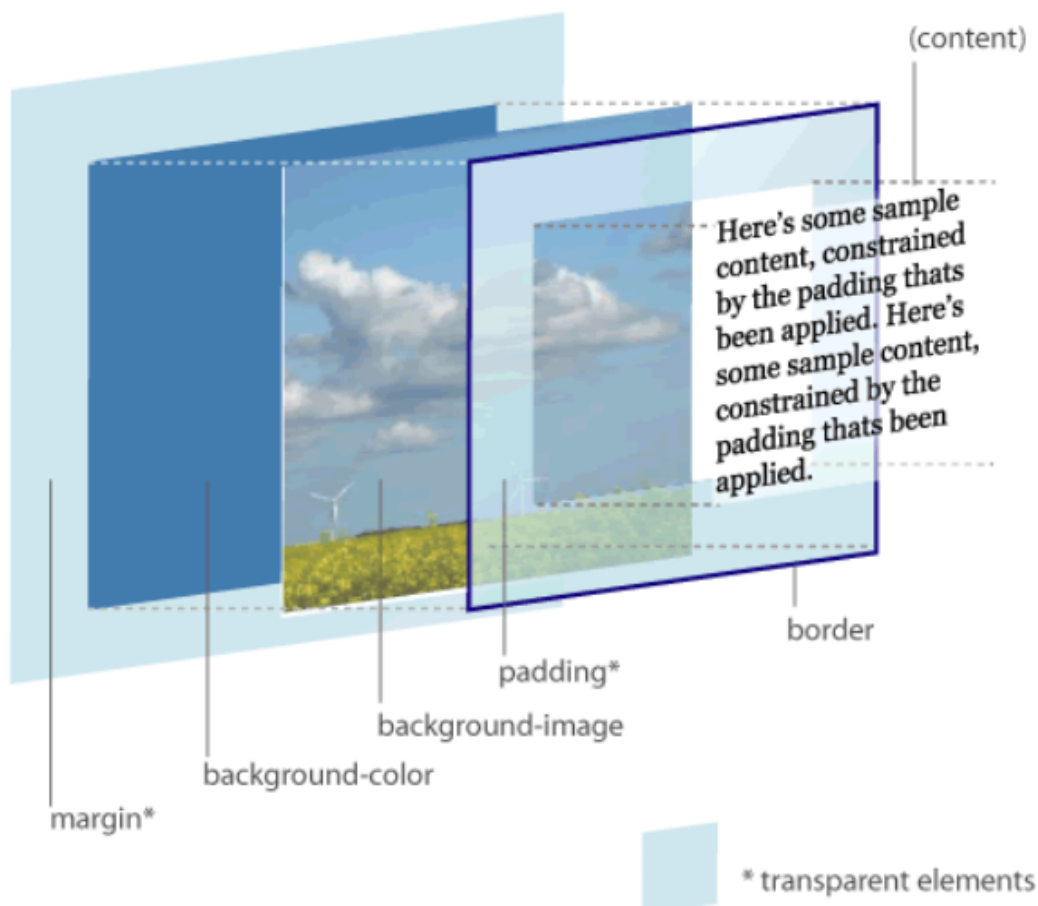


Los navegadores crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características.

2.1.1 PARTES DE UNA CAJA

Cada una de las cajas está formada por seis partes

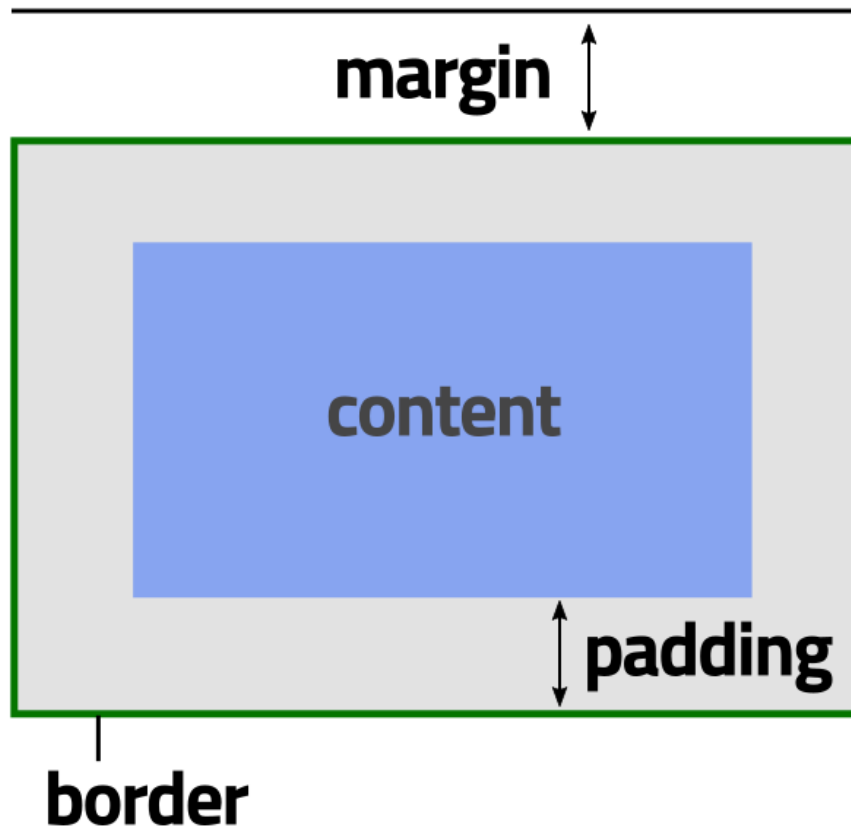
Representación tridimensional del **box model** de CSS



Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes:

- **Contenido** (*content*): se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
- **Relleno** (*padding*): espacio libre opcional existente entre el contenido y el borde.
- **Borde** (*border*): línea que encierra completamente el contenido y su relleno.
- **Imagen de fondo** (*background image*): imagen que se muestra por detrás del contenido y el espacio de relleno.
- **Color de fondo** (*background color*): color que se muestra por detrás del contenido y el espacio de relleno.
- **Margen** (*margin*): separación opcional existente entre la caja y el resto de cajas adyacentes.

2.1.2 REPRESENTACIÓN BÁSICA DE UNA CAJA



Border o Borde es la línea que encierra completamente el contenido y su relleno. En CSS podemos especificar el estilo, ancho y color del borde.

https://www.w3schools.com/css/css_border.asp

```
border-style: dotted; border-style: solid;  
border-width: 20px 10px;  
border-color: red green blue yellow;
```

Padding o Relleno: es el espacio libre opcional entre el contenido y el borde.

https://www.w3schools.com/css/css_padding.asp

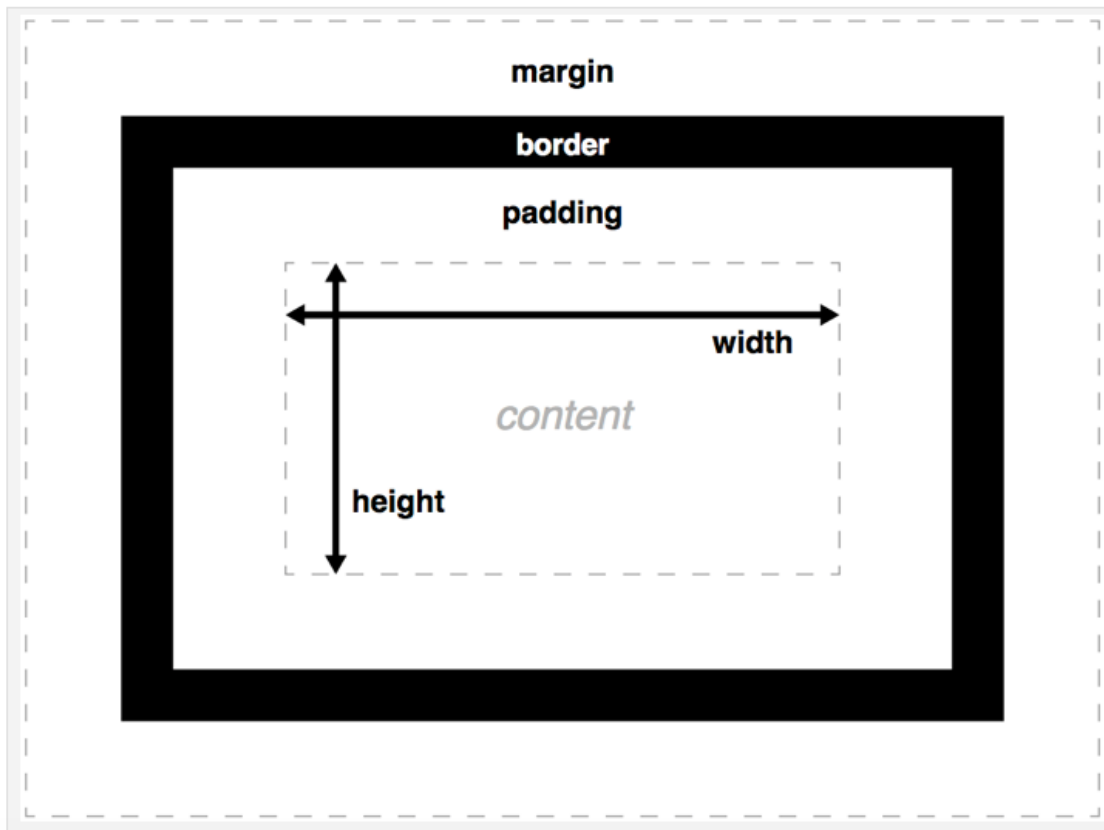
```
padding-top: 25px; padding-right: 50px; padding-bottom: 75px; padding-left: 100px;  
padding: 25px 50px 75px 100px; | padding: 25px; | padding: 25px 50px;
```

Margin o Margen: es la separación opcional existente entre la caja y el resto de cajas adyacentes.

https://www.w3schools.com/css/css_margin.asp

```
margin-top: 100px; margin-bottom: 100px; margin-right: 150px; margin-left: 80px;  
margin: 25px 50px 75px; | margin: 25px 50px; | margin: 50px;
```

2.1.3 DIMENSIONES DE UNA CAJA



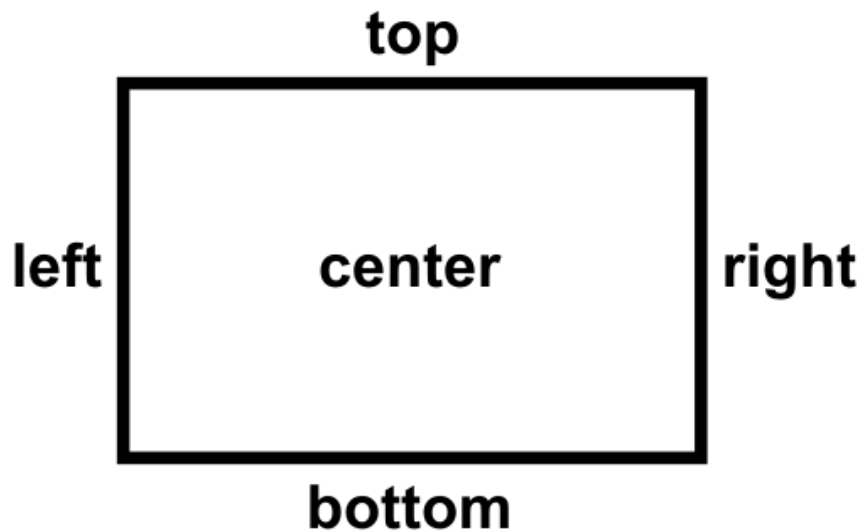
Para dar tamaños específicos a los diferentes elementos de un documento HTML, necesitaremos asignarles valores a las propiedades **width** (ancho) y **height** (alto). Además, ambas tienen propiedades hermanas para establecer el tamaño máximo y mínimo que pueden alcanzar:

Propiedad	Valor	Significado
width:	auto TAMAÑO	Tamaño de ancho de un elemento.
max-width:	none TAMAÑO	Ancho máximo que puede ocupar un elemento.
min-width:	0 TAMAÑO	Ancho mínimo que puede ocupar un elemento.
height:	auto TAMAÑO	Tamaño de alto de un elemento.
max-height:	none TAMAÑO	Alto máximo que puede ocupar un elemento.
min-height:	0 TAMAÑO	Alto mínimo que puede ocupar un elemento.

En el caso de utilizar el valor **auto** en las propiedades anteriores, el navegador se encarga de calcular el ancho o alto necesario, dependiendo del contenido del elemento. Las propiedades **min-width** y **min-height**, por defecto tienen valor 0, mientras que las propiedades **max-width** y **max-height**, por defecto tienen valor none.

2.1.4 ORIENTACIONES DE UNA CAJA

En CSS existen ciertas palabras clave que hacen referencia a una zona u orientación concreta sobre una caja. Son conceptos sencillos y prácticamente lógicos. Son los siguientes:



- **Top**: Se refiere a la parte superior de la caja.
- **Left**: Se refiere a la parte izquierda de la caja.
- **Right**: Se refiere a la parte derecha de la caja.
- **Bottom**: Se refiere a la parte inferior de la caja.
- **Center**: En algunos casos se puede especificar el valor center para referirse a la posición central entre dos extremos.

2.2 HERENCIA Y CASCADA EN CSS

La herencia y la cascada son dos conceptos básicos en CSS. Se deben comprender bien para utilizar CSS.

2.2.1 HERENCIA

La herencia está relacionada con cómo los elementos del etiquetado de HTML heredan propiedades de sus elementos padres (los que los contienen) y los transmiten a sus hijos.

No todas las propiedades CSS son heredadas, porque algunas de ellas no tendría sentido que lo fueran (por ejemplo, los márgenes no se heredan) . La herencia permite que no sea necesario definir y especificar todas las características de todos y cada uno de los tipos de elementos.

Mediante la herencia, por ejemplo, se pueden especificar las propiedades de las fuentes de los elementos html o body y todo el resto de elementos los heredarán.

Forzar la herencia: Mediante la palabra clave **inherit** (heredar) puede forzarse la herencia incluso para propiedades que no se heredan normalmente. (Microsoft Internet Explorer no era compatible con esta palabra clave).

La regla siguiente hace que todos los párrafos hereden todas las propiedades de fondo de sus padres:

```
p {  
    background: inherit;  
}
```

2.2.2 CASCADA

CSS significa *cascading style sheets* (hojas de estilo en cascada). La **cascada** tiene que ver con las declaraciones de CSS que se aplican a un documento y cómo las reglas contradictorias se anulan o no entre ellas. Es el mecanismo que controla el resultado final cuando se aplican varias declaraciones CSS contrapuestas al mismo elemento.

Hay tres conceptos principales que controlan el orden en el que se aplican las declaraciones de CSS:

- Importancia. ¿Dónde se ha especificado la declaración css?
 - Hoja de estilos de navegador
 - Declaraciones normales en hojas de estilo de usuario.
 - Declaraciones importantes en hojas de estilo de usuario.

```
p {  
    background: black !important;  
}
```

- Especificidad. Puede considerarse una medida de cuán específico es el selector de una regla. Por ejemplo en HTML: h1 | h1 class="titulo1" | h1 id="unTitulo"
En Css, las reglas menos específicas a más específicas son: h1 | .titulo1 | #unTitulo

- Orden en las fuentes. Por eso es importante el orden de las reglas en el CSS. De más general a menos general.

2.3 LAS IMÁGENES EN CSS

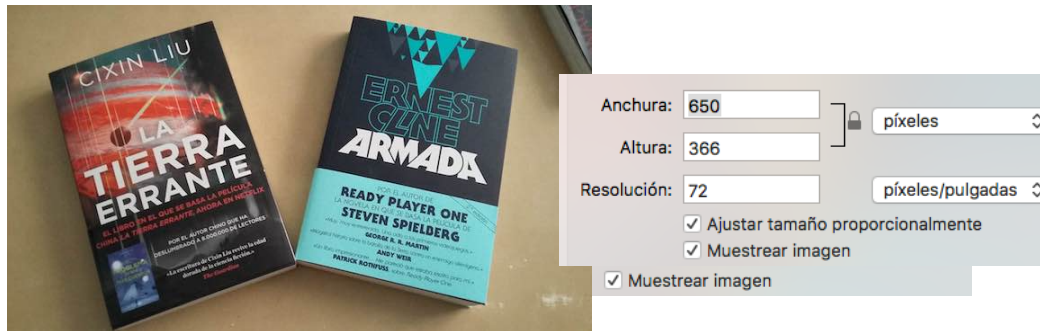
Las **imágenes** son una parte fundamental en el diseño Web. De hecho, la mayoría de las páginas web distribuyen su espacio entre texto e imágenes.

Es necesario que el **largo (height)** y **ancho (width)** de las imágenes no se modifique, para que la imagen no se deforme.

Hay que tener en cuenta que **height y width NO SE HEREDAN**.

La mejor manera, y la más eficiente para los navegadores, es indicar en el HTML el tamaño original de la imagen. Luego, en el CSS, redimensionaremos para ajustarla a la caja donde queremos que aparezca.

1.- Indicar el tamaño original de la imagen en HTML



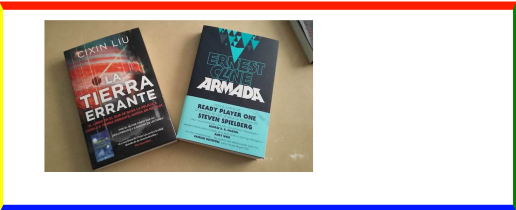


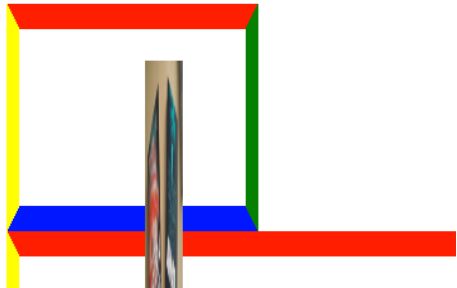

Posibilidades para que la imagen no se deforme:

```
<div class="contenedorImagen">
  
  
  
   → Mantener la proporción
</div>
```

Si se indica el ancho y largo y no coincide con la imagen SE DEFORMA.

2.- Opciones en el CSS indicando o no el tamaño de su caja contenedora y el de la imagen, además de lo indicado en HTML

La mejor opción es indicar que el **width de la imagen sea 100%**, es decir, el tamaño total del contenido de su contenedor.

HTML	CSS	RESULTADO
<pre><div class="contenedorImagen"> </div></pre>	<pre>.contenedorImagen { border-style: solid; border-width: 20px 10px; border-color: red green blue yellow; padding: 25px 50px 75px 100px; /* width: 30px; */ }</pre>	
<pre><div class="contenedorImagen"> </div></pre>	<pre>.contenedorImagen { border-style: solid; border-width: 20px 10px; border-color: red green blue yellow; padding: 25px 50px 75px 100px; width: 30px; }</pre>	 <p>Desbordamiento imagen en width pero caja div con height que contenga la imagen.</p>
<pre><div class="contenedorImagen"> </div></pre>	<pre>.contenedorImagen { border-style: solid; border-width: 20px 10px; border-color: red green blue yellow; padding: 25px 50px 75px 100px; width: 30px; height: 40px; }</pre>	 <p>Desbordamiento imagen en width y height ya que está definido en caja div.</p>
<pre><div class="contenedorImagen"> </div></pre>	<pre>.contenedorImagen { border-style: solid; border-width: 20px 10px; border-color: red green blue yellow; padding: 25px 50px 75px 100px; width: 30px; height: 40px; } img.ancho { width: 100%; }</pre>	 <p>Desproporción de la imagen para mantener width y height que se indica en HTML.</p>
<pre><div class="contenedorImagen"> </div></pre>	<pre>.contenedorImagen { ... width: 30px; height: 40px; } img.ancho { width: 100%; }</pre>	 <p>Cuadra la imagen en la cada div, manteniendo proporción y con los borders decididos.</p>

Centrar imágenes en contenedores div:

Las imágenes no se pueden centrar en las cajas contenedoras como elementos de línea (**text-align: center;**) como los párrafos, con CSS ya que son elementos de bloque.

El código CSS para centrar la imagen en el contenedor es:

```
img {  
  display: block;    // se indica que se trate como un elemento de bloque  
  margin: auto;      // el navegador ajusta los márgenes de manera automática  
}
```

2.4 POSICIONAMIENTO EN CSS

Los navegadores crean y posicionan de forma automática todas las cajas que forman cada página HTML. No obstante, CSS permite al diseñador modificar la posición en la que se muestra cada caja.

El estándar de CSS define cinco modelos diferentes para posicionar una caja.

1.- Posicionamiento predefinido (estático o normal):

De forma predefinida los elementos de bloque aparecen unos debajo de otros según el flujo en que los pongamos en el documento. Si asignamos una altura o anchura a uno, los que pongamos en su interior estarán limitados a ese tamaño. Es el posicionamiento que utilizan los navegadores si no se indica lo contrario.

2.- Posicionamiento relativo

Podemos especificar desplazamiento a una caja con respecto a su posición original en el flujo de ejecución (predefinida).

Utilizamos la propiedad **position** más las propiedades **top**, **left**, **right** y **bottom** que aplican el desplazamiento:

```
div .clase2 {  
  position: relative;  
  left: 15px;  
}
```

Por ejemplo, desplazamos el elemento 15px respecto a su borde izquierdo, de modo que observaremos un desplazamiento hacia la derecha.

Este tipo de posicionamiento no modifica el posicionamiento de las demás cajas a su alrededor, pero sí puede provocar solapamientos.

Ejemplo de prácticas con posicionamiento relativo.

Diferenciar entre posicionamiento relativo y márgenes. Para ciertos casos, el resultado es el mismo.

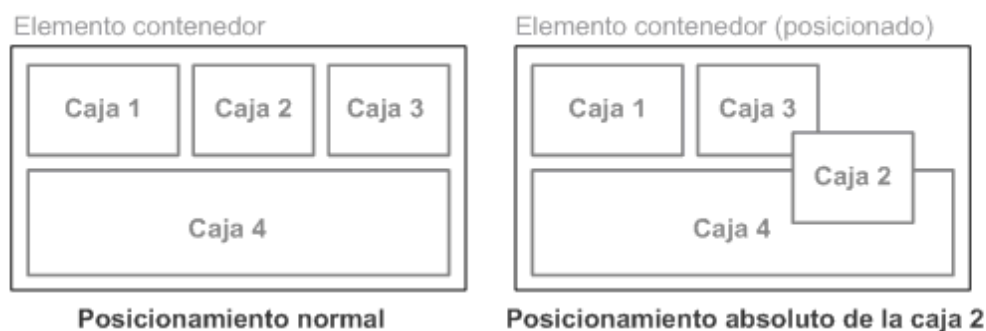
- margin-top: 20px; Afecta al tamaño global de la caja. Si el navegador tiene que buscar espacio para situar esta caja tendrá en cuenta el margen asignado.
- position relative; top: 20px; Afecta solo a la posición pero el tamaño de la caja es el mismo

3.- Posicionamiento absoluto

El posicionamiento absoluto se emplea para establecer de forma exacta la posición en la que se muestra la caja de un elemento. La nueva posición de la caja se indica mediante las propiedades **top**, **left**, **right** y **bottom**. La interpretación de los valores de estas propiedades es mucho más compleja que en el posicionamiento relativo, ya que en este caso **dependen del posicionamiento del elemento contenedor que esté posicionado**, y si no existe, la referencia será la ventana del navegador.

Cuando una caja se posiciona de forma absoluta, el resto de elementos de la página se ven afectados y modifican su posición, tratando los demás elementos de rellenar el hueco abandonado. Al igual que en el posicionamiento relativo, cuando se posiciona de forma absoluta una caja es probable que se produzcan solapamientos con otras cajas.

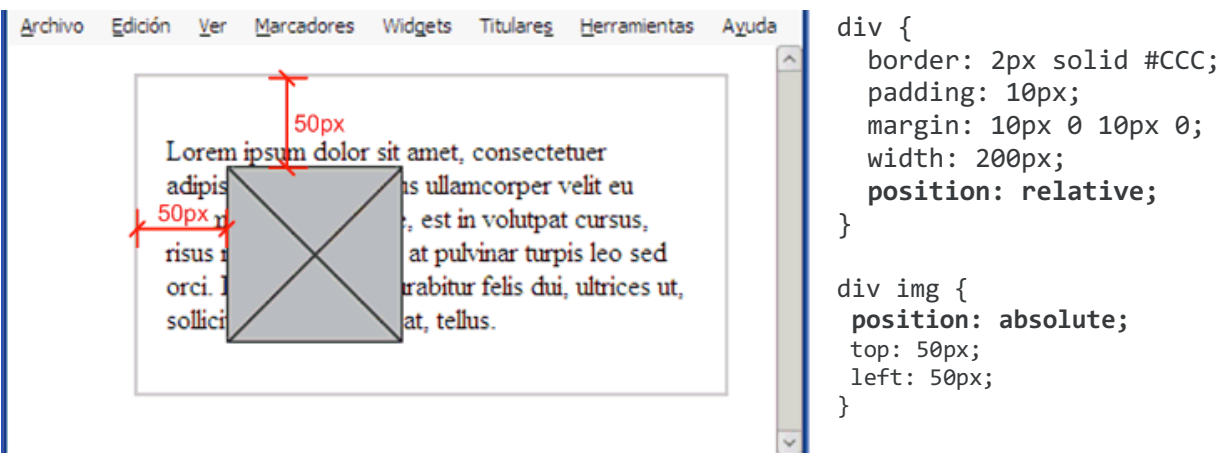
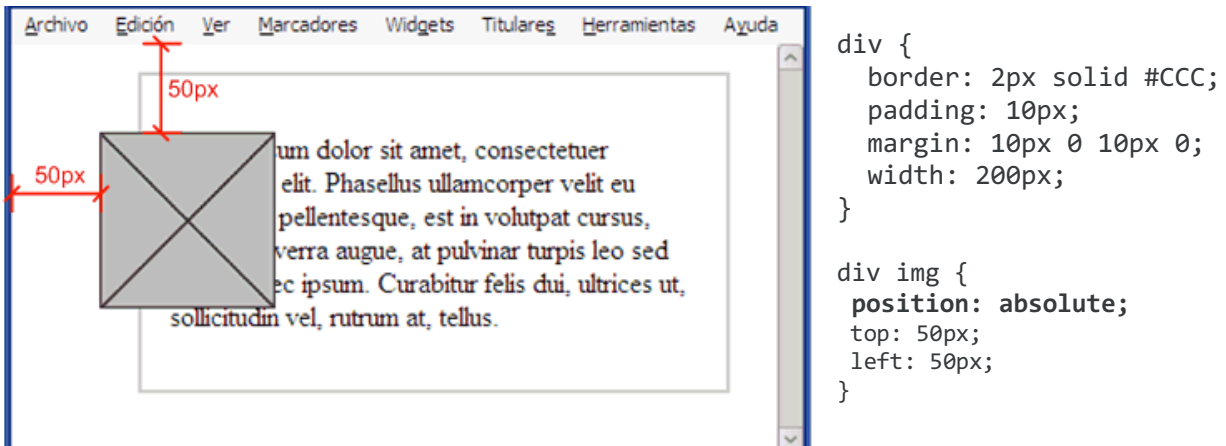
```
div .caja2 {  
    position: absolute;  
    left: 15px;  
}
```



La caja 2 está posicionada de forma absoluta, lo que provoca que el resto de elementos de la página modifiquen su posición. En concreto, la caja 3 deja su lugar original y pasa a ocupar el hueco dejado por la caja 2.

Las cajas posicionadas de forma absoluta "**salen del flujo normal de la página**", lo que provoca que el resto de elementos de la página se muevan y en ocasiones, ocupen la posición original en la que se encontraba la caja.

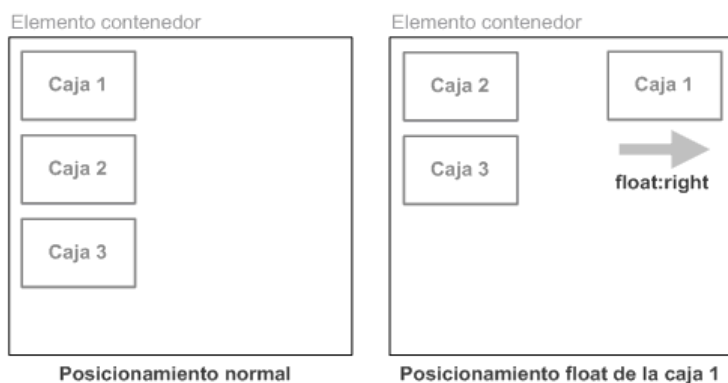
Otro ejemplo:



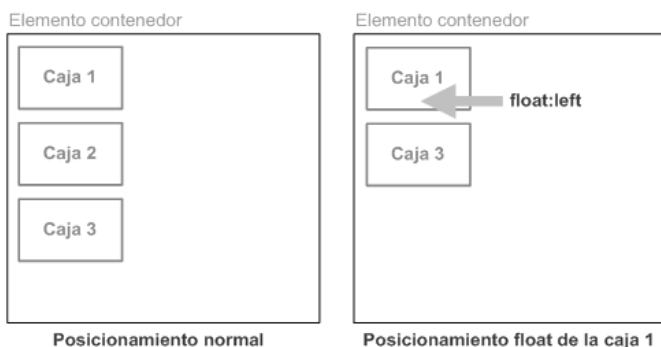
Por tanto, si se quiere posicionar un elemento de forma absoluta respecto de su elemento contenedor, es imprescindible posicionar este último. Para ello, sólo es necesario añadir la propiedad `position: relative`.

4.- Posicionamiento flotante

El posicionamiento flotante es el más utilizado. Cuando una caja se posiciona con el modelo de posicionamiento flotante, automáticamente se convierte en una *caja flotante*, lo que significa que se desplaza hasta la zona más a la izquierda o más a la derecha de la posición en la que originalmente se encontraba.

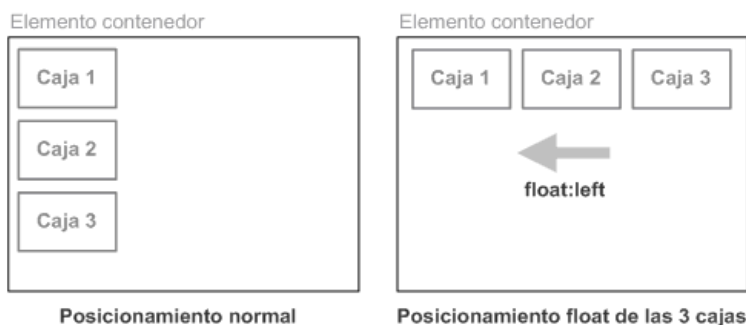


Cuando se posiciona una caja de forma flotante deja de pertenecer al flujo normal de la página, lo que significa que el resto de cajas ocupan el lugar dejado por la caja flotante. La **caja flotante caja 1**, se posiciona lo más a la **derecha** posible de la posición en la que se encontraba originalmente.

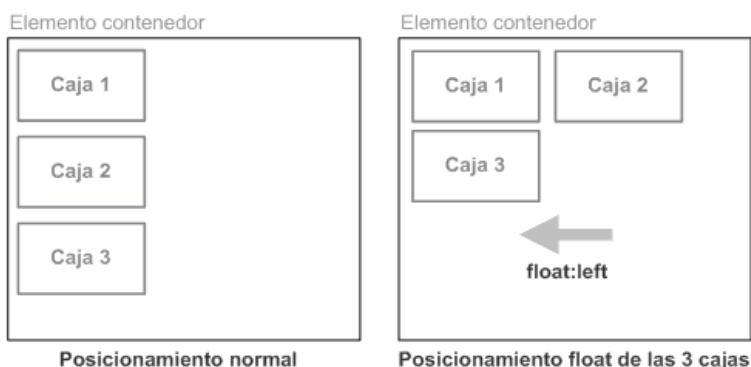


La **caja 1** es de tipo flotante, por lo que *desaparece del flujo normal* de la página y el resto de cajas ocupan su lugar. El resultado es que la caja 2 ahora se muestra donde estaba la caja 1 y la caja 3 se muestra donde estaba la caja 2.

Al mismo tiempo, la **caja 1 se desplaza todo lo posible hacia la izquierda** de la posición en la que se encontraba. El resultado es que la caja 1 se muestra encima de la nueva posición de la caja 2 y tapa todos sus contenidos.



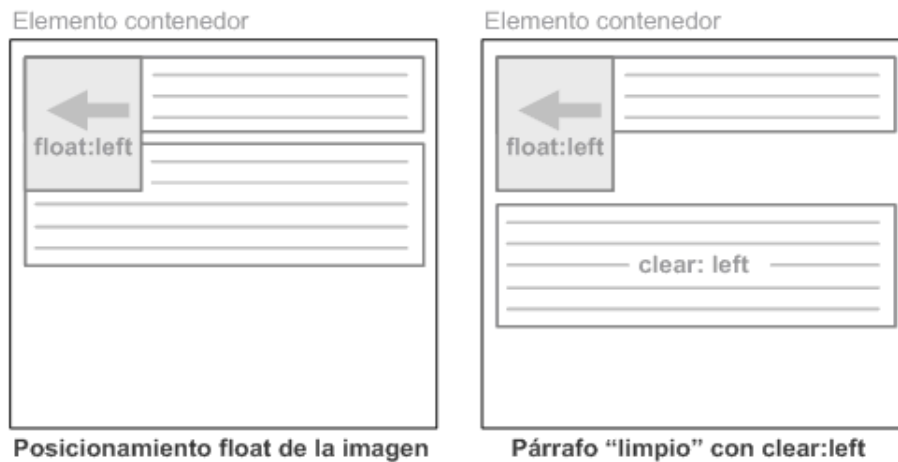
Todas las cajas son flotantes. Las cajas no se superponen entre sí porque las cajas flotantes tienen en cuenta las otras cajas flotantes existentes. Como la caja 1 ya estaba posicionada lo más a la izquierda posible, la caja 2 sólo puede colocarse al lado del borde derecho de la caja 1, que es el sitio más a la izquierda posible respecto de la zona en la que se encontraba.



Si no existe sitio en la línea actual, la caja flotante baja a la línea inferior hasta que encuentra el sitio necesario para mostrarse lo más a la izquierda o lo más a la derecha posible en esa nueva línea.

Uno de los principales motivos para la creación del posicionamiento **float** fue precisamente la posibilidad de colocar imágenes alrededor de las cuales fluye el texto.

CSS permite controlar la forma en la que **los contenidos fluyen alrededor de los contenidos posicionados mediante float**.



La propiedad **clear** permite modificar el comportamiento por defecto del posicionamiento flotante para forzar a un elemento a mostrarse debajo de cualquier caja flotante. La regla CSS que se aplica al segundo párrafo del ejemplo anterior es la siguiente:

```
clear: left;
```