Capítulo 6

Posicionamiento y visualización

Posicionamiento y visualización

- Los navegadores crean una caja para representar a cada elemento de la página **HTML**.

- Los factores que se tienen en cuenta para generar cada caja son:
  - Las propiedades **width** y **height** de la caja.
  - El *tipo* de cada elemento **HTML** (elemento de bloque o elemento en línea).
  - *Posicionamiento* de la caja (normal, relativo, absoluto, fijo o flotante).
  - Las *relaciones* entre elementos (dónde se encuentra cada elemento, elementos descendientes, etc.)
  - Otro tipo de información, como por ejemplo el *tamaño* de las imágenes y el tamaño de la ventana del navegador.

Tipos de elementos

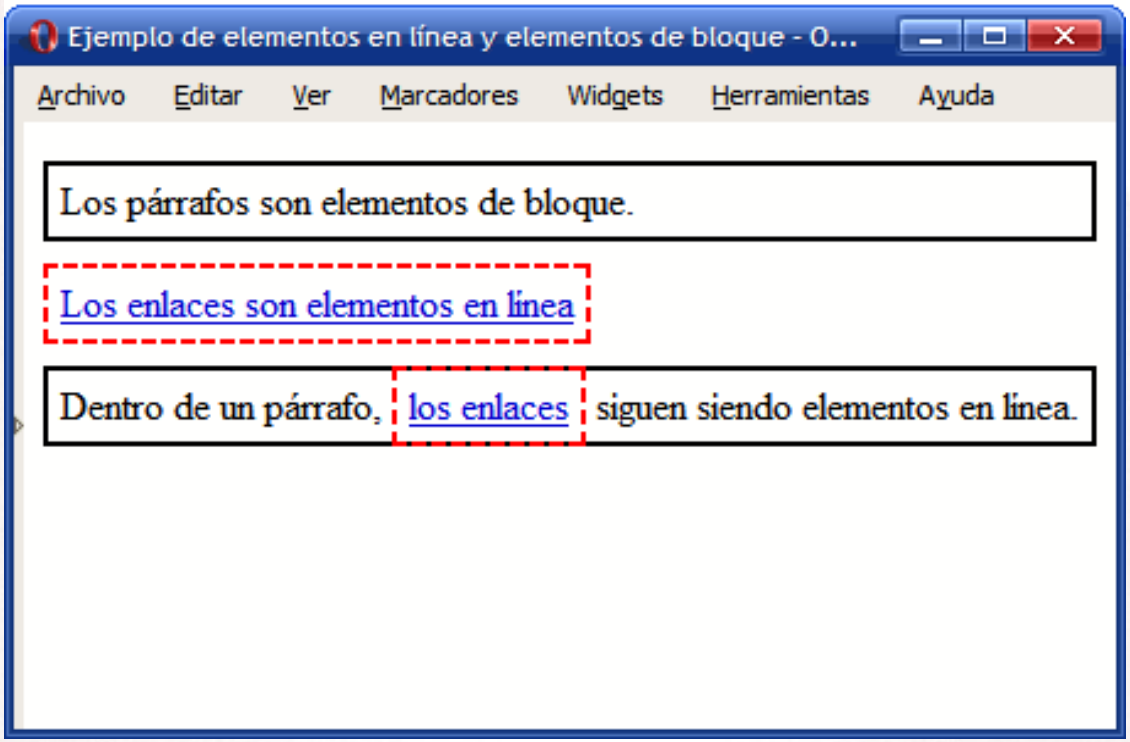
- HTML clasifica a todos sus elementos en dos grandes grupos:
  - elementos en línea y
  - elementos de bloque.

- Los *elementos de bloque* siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea.

- Los *elementos en línea* no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.

Tipos de elementos

- El tipo de un elemento influye de forma decisiva en la caja que el navegador crea para mostrarlo.



Tipos de elementos

- Los elementos en línea definidos por HTML son: a, abbr, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, 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, isindex, menu, noscript, ol, p, pre, table, ul.

Posicionamiento

- Los navegadores crean y posicionan de forma *automática* todas las cajas que forman cada página HTML.

- **CSS** define *cinco modelos diferentes* para posicionar una caja:
  - **Posicionamiento normal o estático:** se trata del posicionamiento que utilizan los navegadores por defecto.
  - **Posicionamiento relativo:** consiste en posicionar una caja según el posicionamiento normal y después desplazarla respecto de su posición original.

Posicionamiento

...

- **Posicionamiento absoluto:** la posición de una caja se establece de forma absoluta respecto de su elemento contenedor y el resto de elementos de la página ignoran la nueva posición del elemento.
- **Posicionamiento fijo:** variante del posicionamiento absoluto que convierte una caja en un elemento inamovible, de forma que su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.
- **Posicionamiento flotante:** desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.

Posicionamiento

- El posicionamiento de una caja se establece mediante la propiedad **position:**

position	Posicionamiento
Valores	`static` | `relative` | `absolute` | `fixed` | `inherit`
Se aplica a	Todos los elementos
Valor inicial	static
Descripción	Selecciona el posicionamiento con el que se mostrará el elemento

Posicionamiento

- **CSS** define cuatro propiedades llamadas **top**, **right**, **bottom** y **left** para controlar el desplazamiento de las cajas posicionadas:

top **right** **bottom** **left**	Desplazamiento superior Desplazamiento lateral derecho Desplazamiento inferior Desplazamiento lateral izquierdo
Valores	`<medida>` | `<porcentaje>` | `auto` | `inherit`
Se aplica a	Todos los elementos posicionados
Valor inicial	auto
Descripción	Indican el desplazamiento horizontal y vertical del elemento respecto de su posición original

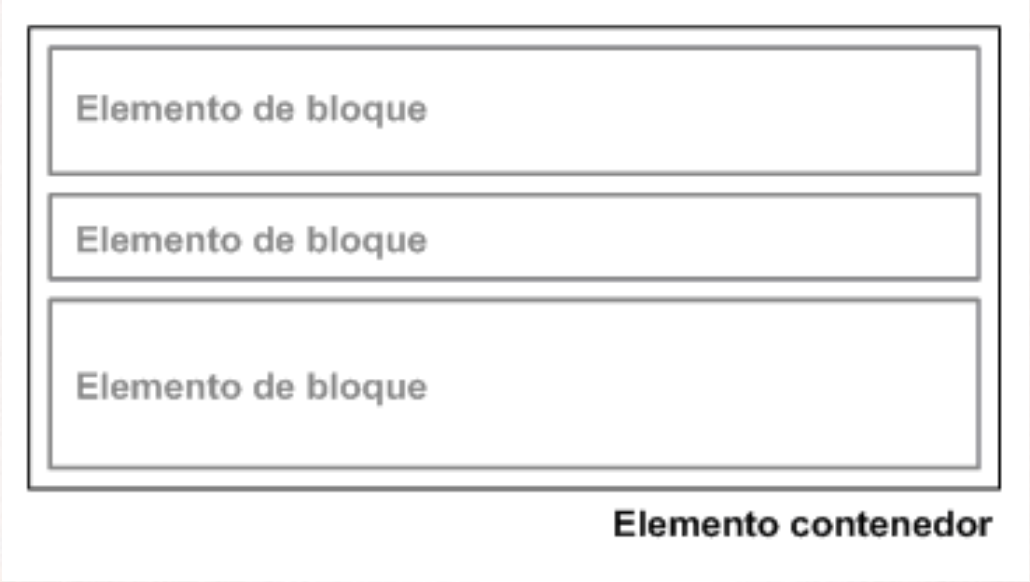
Posicionamiento normal

- El *posicionamiento normal* o *estático* es el modelo que utilizan *por defecto* los navegadores para mostrar los elementos de las páginas.

- Ninguna caja se desplaza respecto de su posición original, *sólo se tiene en cuenta si el elemento es de bloque o en línea*.

Posicionamiento normal

- Las cajas se muestran una debajo de otra comenzando desde el principio del elemento contenedor.
- La distancia entre las cajas se controla mediante los **márgenes** verticales.



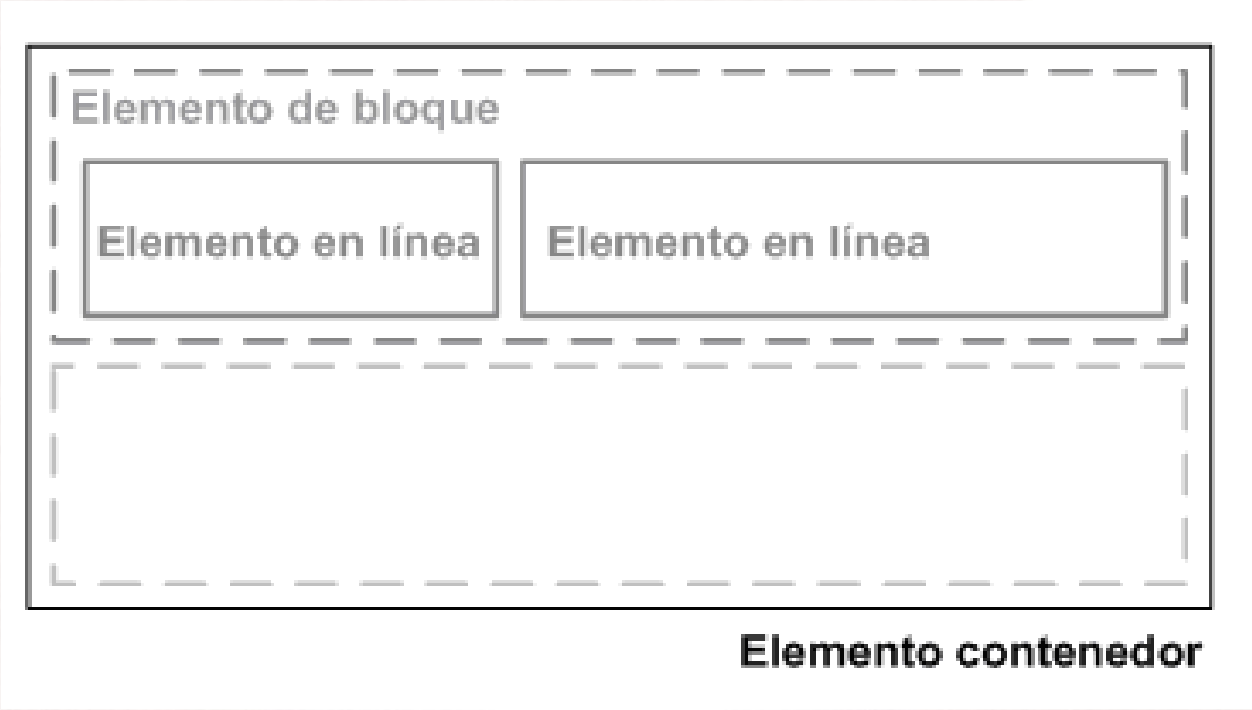
Posicionamiento normal

- La *anchura* de los elementos de bloque está *limitada* a la anchura de su elemento contenedor,
  - en algunos casos sus contenidos pueden **desbordar** el espacio disponible.

- Los elementos en línea forman los *"contextos de formato en línea"*.
  - En este tipo de contextos, las cajas se muestran una detrás de otra de forma horizontal comenzando desde la posición más a la izquierda de su elemento contenedor.

Posicionamiento normal

- La distancia entre las cajas se controla mediante los márgenes laterales.



Posicionamiento normal

- Si las cajas en línea ocupan más espacio del disponible en su propia línea, el resto de cajas se muestran en las líneas inferiores.

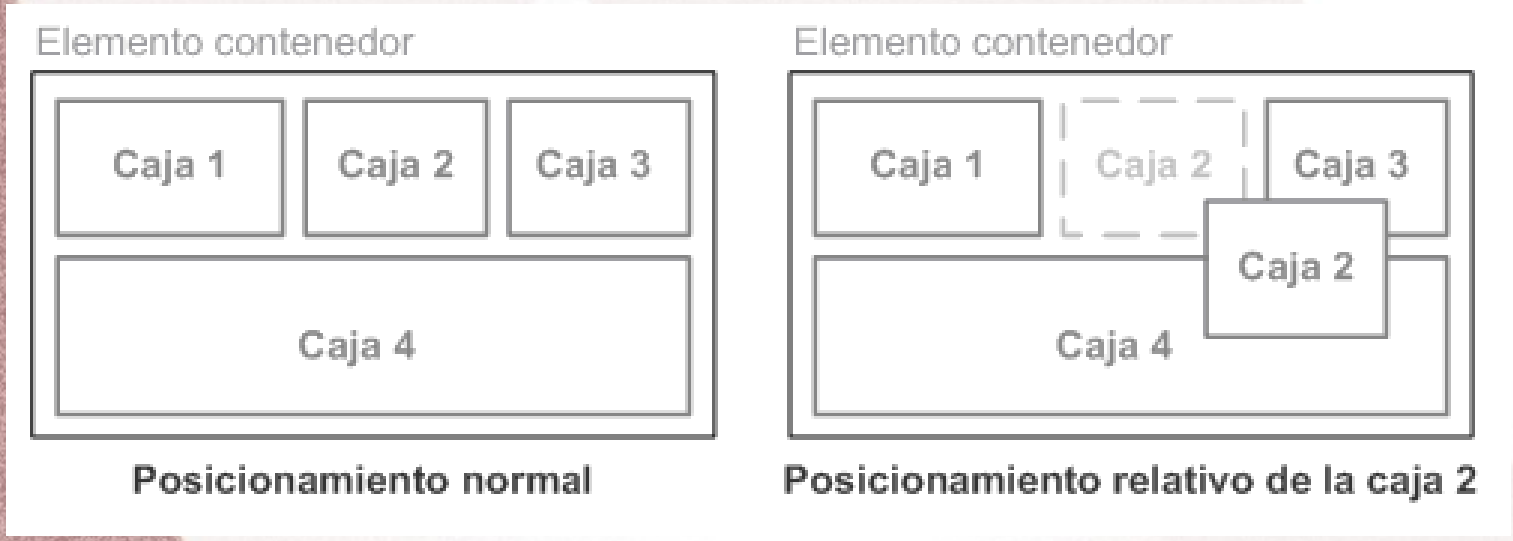
Posicionamiento relativo

- El *posicionamiento relativo* permite desplazar una caja respecto de su posición original establecida mediante el *posicionamiento normal*.

- El *desplazamiento* de la caja se controla con las propiedades **top**, **right**, **bottom** y **left**.

Posicionamiento relativo

- El *desplazamiento* de una caja *no afecta al resto de cajas adyacentes*.

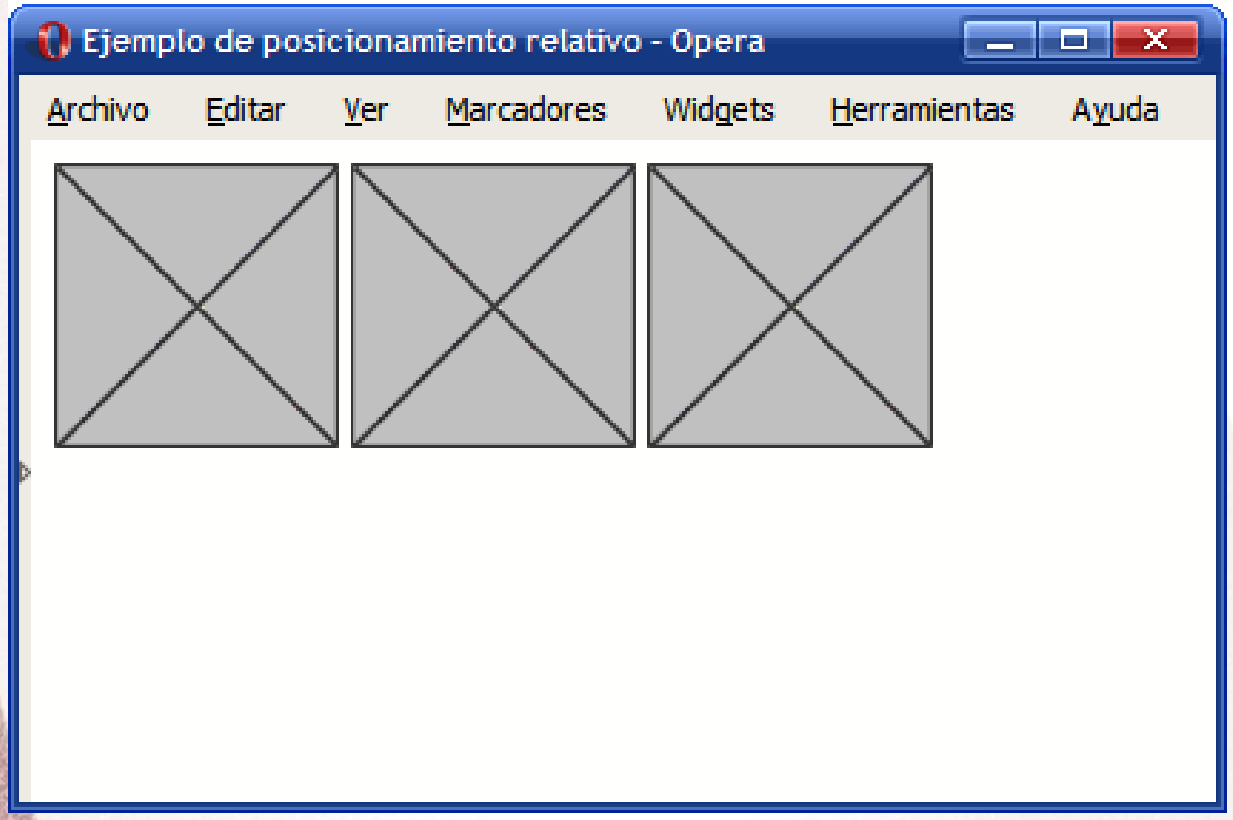


Posicionamiento relativo

- Si se utilizan valores negativos en estas propiedades, su efecto es justamente el *inverso*.
- La propiedad **direction** permite establecer la dirección del texto.
  - Valores posibles: ltr o rtl

Posicionamiento relativo

- El siguiente ejemplo muestra tres imágenes posicionadas de forma normal:



Posicionamiento relativo

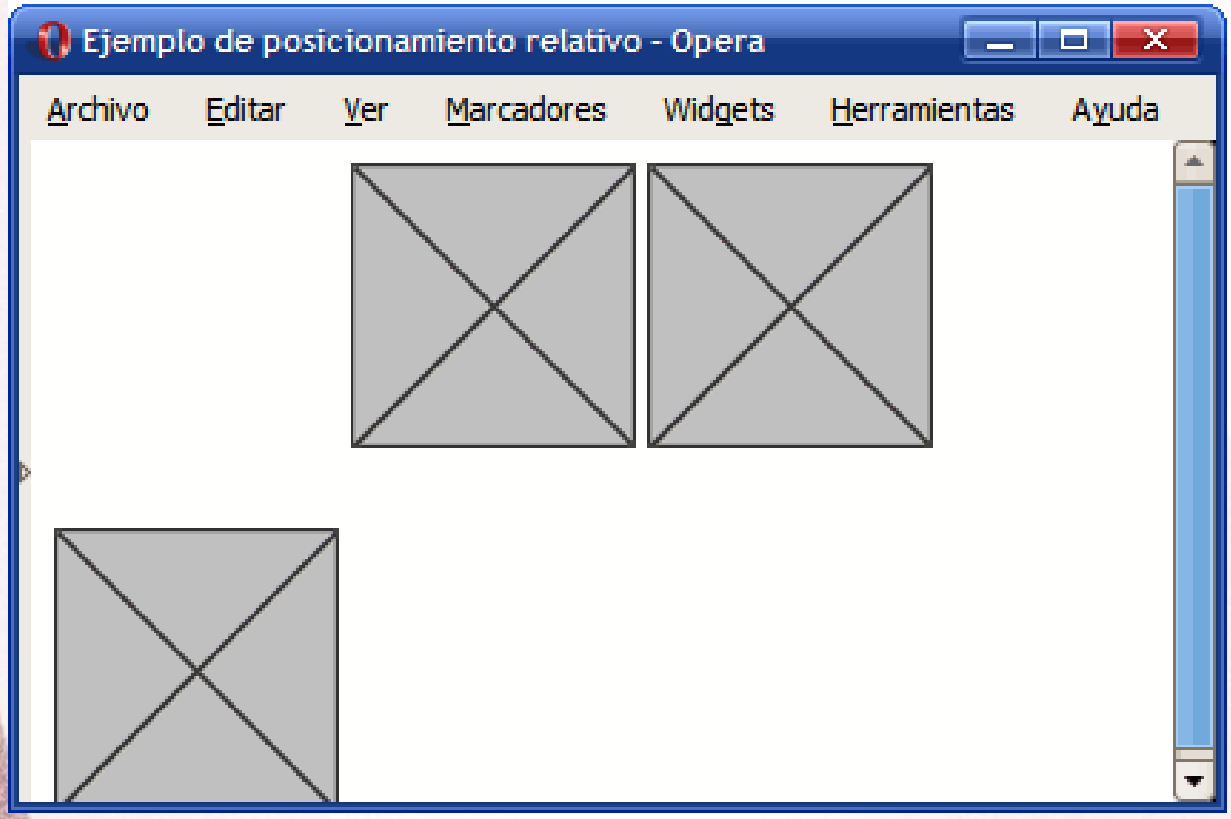
- Aplicando el posicionamiento relativo, se desplaza la primera imagen de forma descendente:

```
img.desplazada {
position: relative;
top: 8em;
}
```

```
<img class="desplazada" src="imagenes/imagen.png" alt="Imagen genérica" />
<img src="imagenes/imagen.png" alt="Imagen genérica" />
<img src="imagenes/imagen.png" alt="Imagen genérica" />
```

Posicionamiento relativo

- El aspecto que muestran ahora las imágenes es el siguiente:



Posicionamiento relativo

- El *resto de imágenes* *no varían su posición.*

- *Problema:* se pueden producir **solapamientos** con otros elementos de la página.

Posicionamiento absoluto

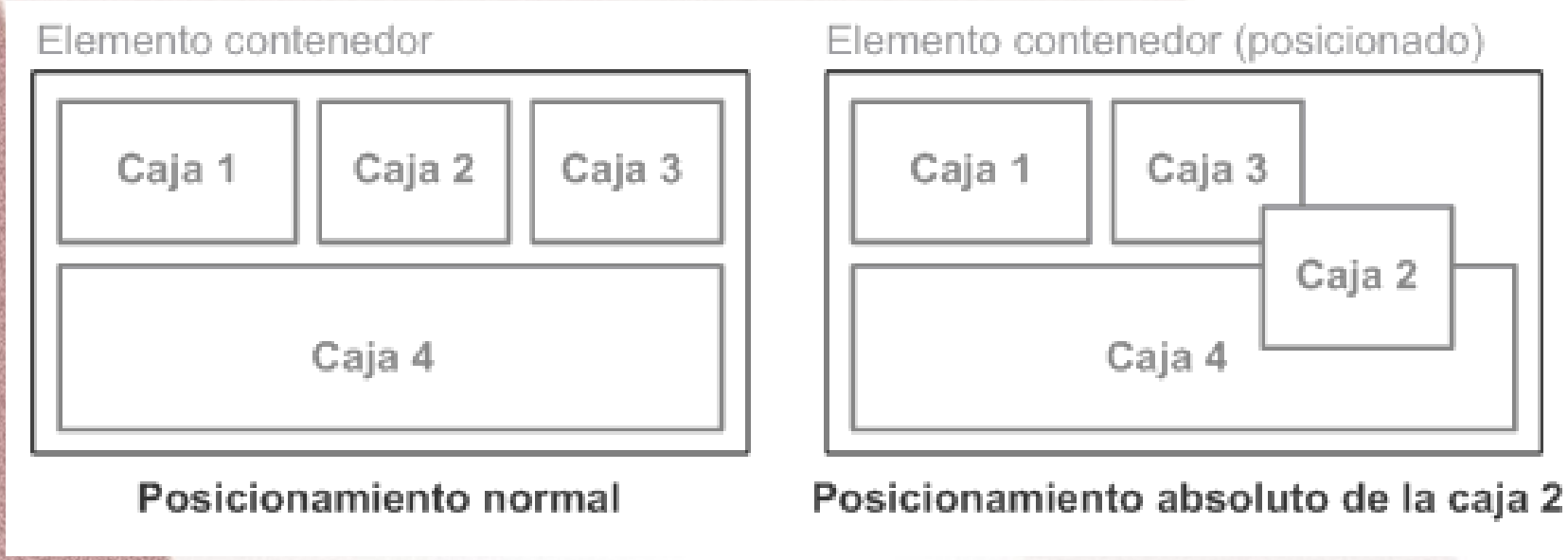
- El posicionamiento absoluto se emplea para establecer de forma precisa la posición en la que se muestra la caja de un elemento.

- La nueva posición de la caja se indica mediante las propiedades **top**, **right**, **bottom** y **left**.

- En este caso depende del posicionamiento del elemento contenedor.

Posicionamiento absoluto

- **Ejemplo**: se posiciona de forma absoluta la caja 2:



Posicionamiento absoluto

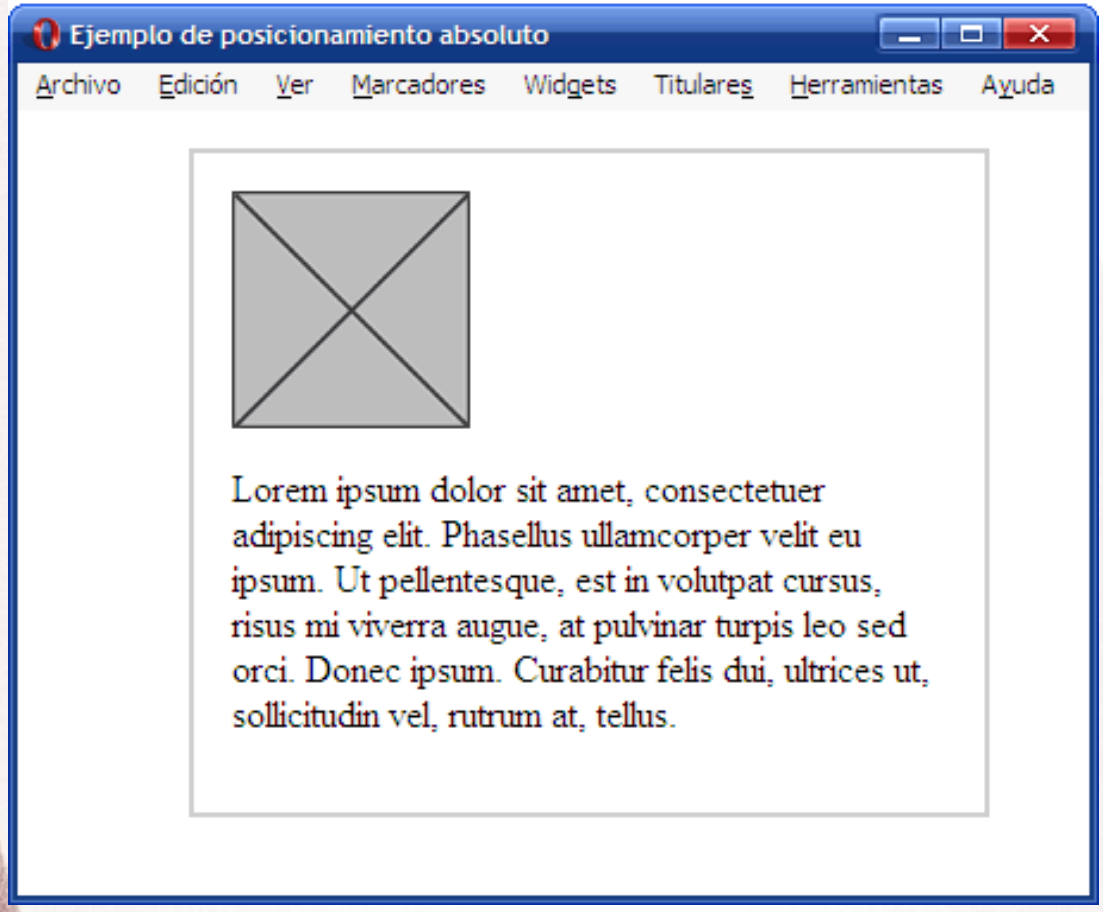
- Determinar el origen de coordenadas a partir del cual se desplaza una caja posicionada de forma absoluta se compone de los siguientes pasos:
  - Se buscan todos los elementos contenedores de la caja hasta llegar al elemento **<body>** de la página.
  - Se recorren todos los elementos contenedores empezando por el más cercano a la caja y llegando hasta el **<body>**
  - De todos ellos, el navegador se queda con el primer elemento contenedor que esté posicionado de cualquier forma diferente a *position: static*
  - La esquina superior izquierda de ese elemento contenedor posicionado es el origen de coordenadas.

Posicionamiento absoluto

- Una vez obtenido el origen de coordenadas, se interpretan los valores de las propiedades **top**, **right**, **bottom** y **left** respecto a ese origen y se desplaza la caja hasta su nueva posición.

Posicionamiento absoluto

- Ejemplo:



Posicionamiento absoluto

- El código HTML y CSS de la página original:

```
div {
    border: 2px solid #CCC;
    padding: 1em;
    margin: 1em 0 1em 4em;
    width: 300px;
}

<div>
    <img src="imagenes/imagen.png" alt="Imagen genérica" />
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Phasellus ullamcorper velit eu ipsum. Ut pellentesque, est in volutpat cursus,
risus mi viverra augue, at pulvinar turpis leo sed orci. Donec ipsum. Curabitur
    felis dui, ultrices ut, sollicitudin vel, rutrum at, tellus.</p>
</div>
```

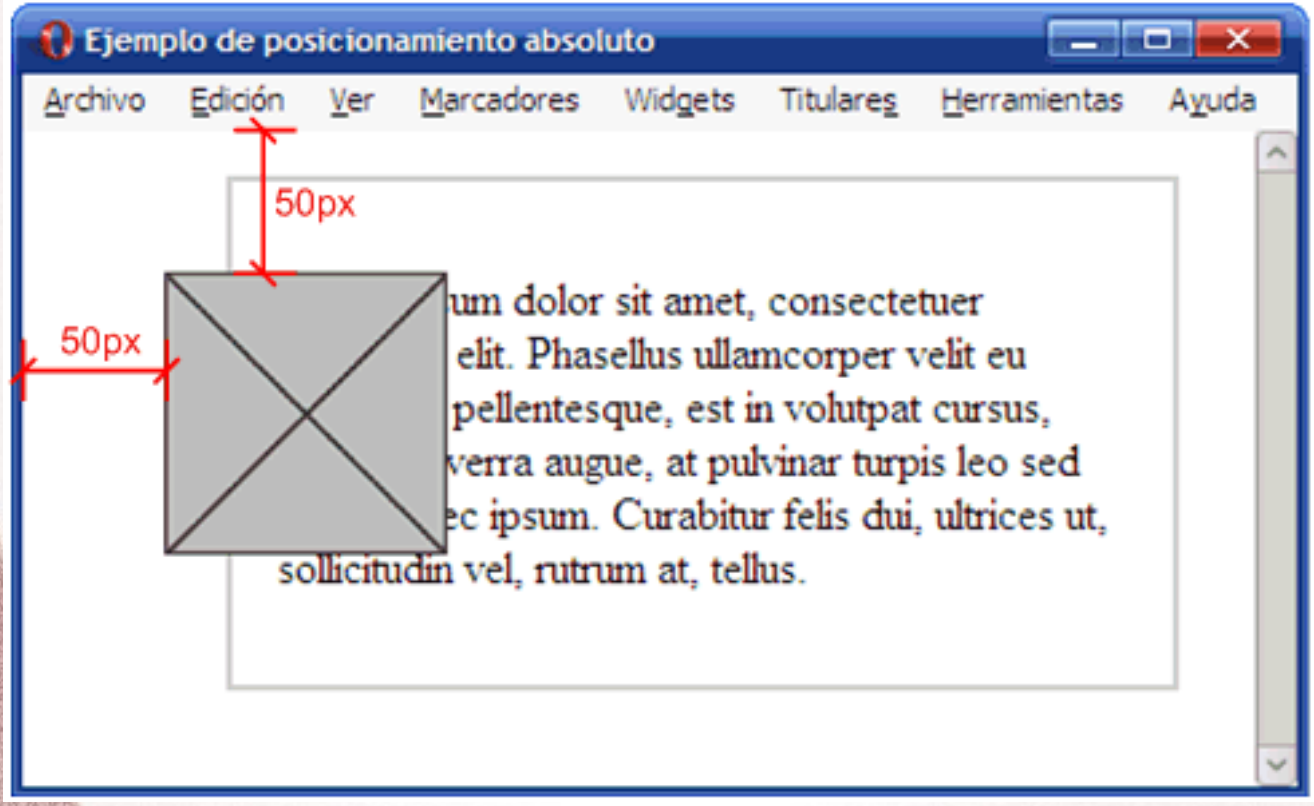
Posicionamiento absoluto

- Se posiciona de forma absoluta la imagen mediante la propiedad **position** y se indica su nueva posición mediante las propiedades **top** y **left**:

```
div img {
    position: absolute;
    top: 50px;
    left: 50px;
}
```

Posicionamiento absoluto

- La imagen posicionada de forma absoluta toma como origen de coordenadas la esquina superior izquierda de la página:



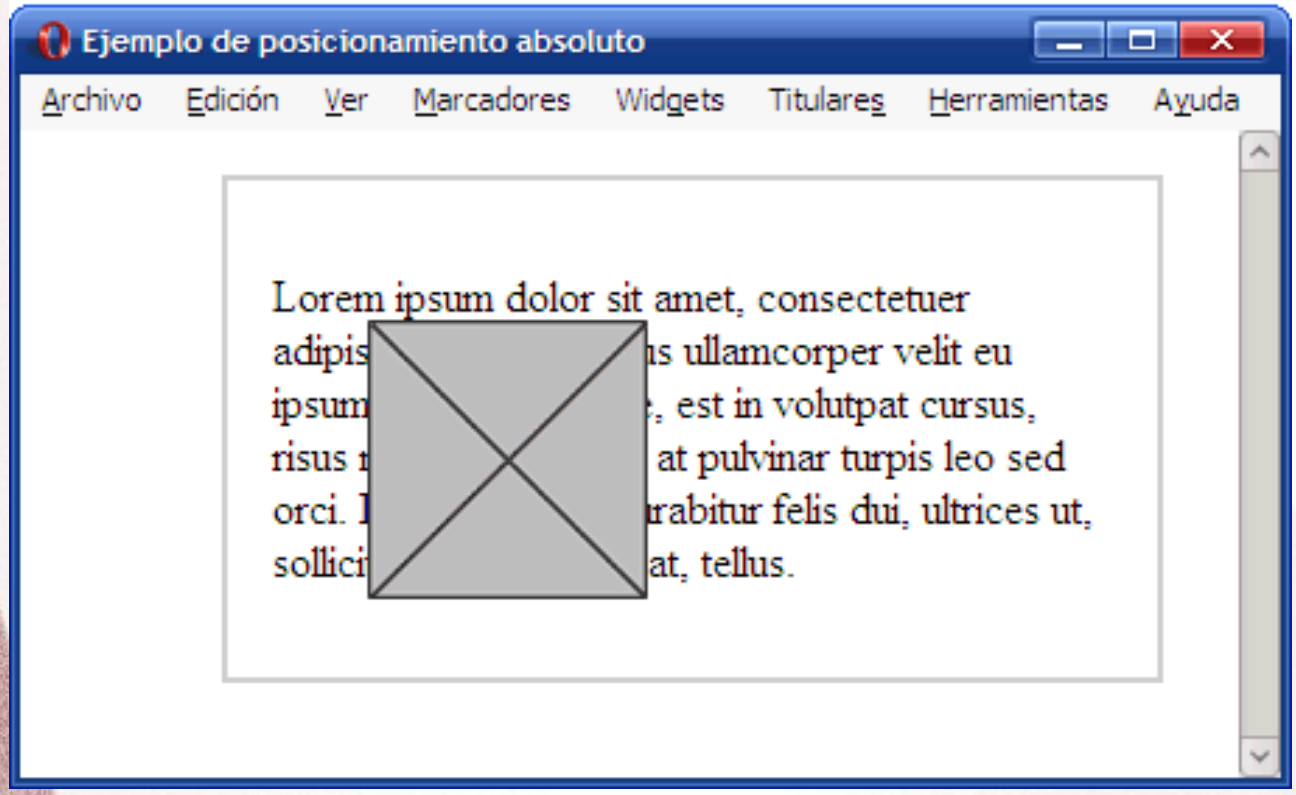
Posicionamiento absoluto

- La única propiedad añadida al ** ** es *position: relative* por lo que el elemento contenedor se posiciona pero no se desplaza respecto de su posición original:

```
div {
    border: 2px solid #CCC;
    padding: 1em;
    margin: 1em 0 1em 4em;
    width: 300px;
    position: relative;
}
div img {
    position: absolute;
    top: 50px;
    left: 50px;
}
```

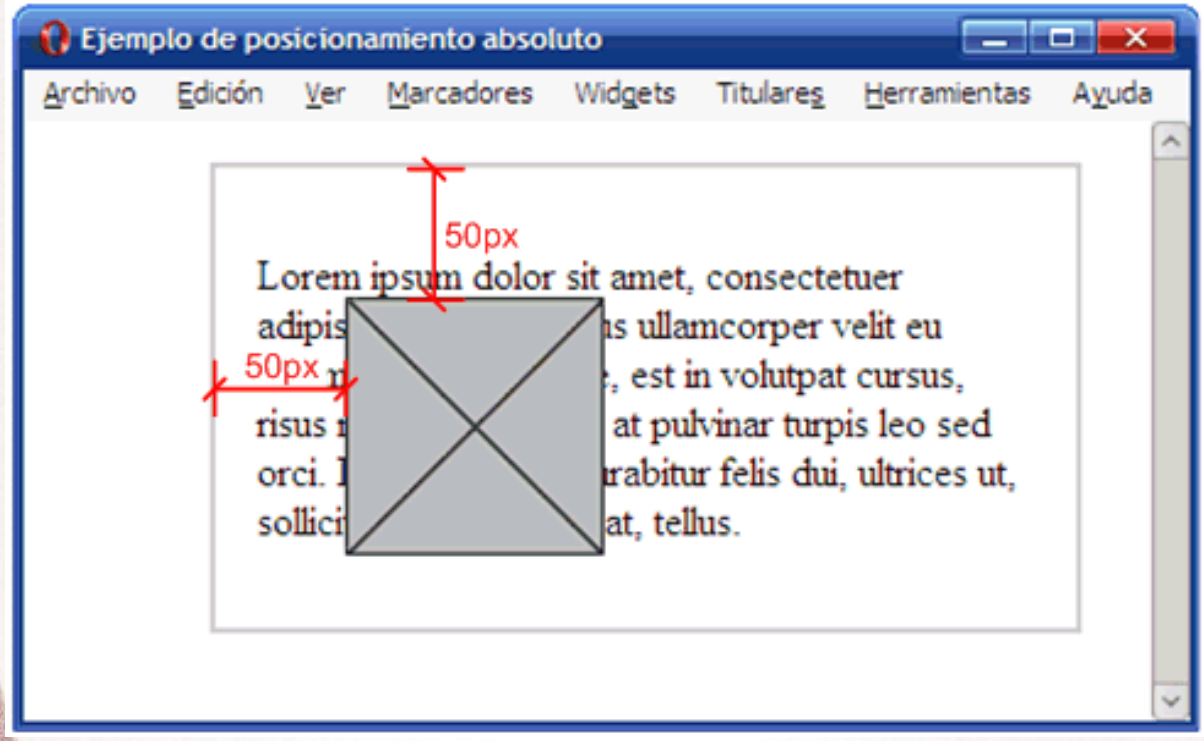
Posicionamiento absoluto

- La siguiente imagen muestra el resultado obtenido:



Posicionamiento absoluto

- En este caso, el origen de coordenadas para determinar la nueva posición de la imagen corresponde a la esquina superior izquierda del elemento ** **:



Posicionamiento absoluto

- Si se quiere posicionar un elemento de forma absoluta respecto de su elemento contenedor, es imprescindible posicionar el elemento contenedor.

- Para ello, sólo es necesario añadir la propiedad *position: relative*, por lo que no es obligatorio desplazar el elemento contenedor respecto de su posición original.

- Ejemplos:
  - https://codepen.io/ioc-daw-m09/pen/xVPpXX

Posicionamiento fijo

- position: fixed
- Los elementos a los cuales se les posiciona con position: fixed también están fuera del flujo normal de la página. Sin embargo, no se debe de confundir con los elementos que están posicionados con position: absolute.

- A diferencia de estos últimos, los elementos con position: fixed toman como referencia la ventana del navegador y no respetan el tener un contenedor padre que esté posicionado. Además, al hacer scroll en la página, el elemento que esté posicionado como position: fixed seguirá en la misma posición respecto a la ventana del navegador aunque el scroll haya desplazado la página hacia abajo.

Posicionamiento fijo

- position: fixed
- Los elementos a los cuales se les posiciona con position: fixed también están fuera del flujo normal de la página. Sin embargo, no se debe de confundir con los elementos que están posicionados con position: absolute.

- A diferencia de estos últimos, los elementos con position: fixed toman como referencia la ventana del navegador y no respetan el tener un contenedor padre que esté posicionado. Además, al hacer scroll en la página, el elemento que esté posicionado como position: fixed seguirá en la misma posición respecto a la ventana del navegador aunque el scroll haya desplazado la página hacia abajo.

Posicionamiento fijo

- Cuando una caja se posiciona de forma fija, la forma de obtener el origen de coordenadas para interpretar su desplazamiento es idéntica al posicionamiento absoluto.

- Si el usuario no mueve la página HTML en la ventana del navegador (scroll), no existe ninguna diferencia entre estos dos modelos de posicionamiento.

Posicionamiento fijo

- Posición inamovible dentro de la ventana del navegador.
- Las cajas no modifican su posición ni aunque el usuario suba o baje la página en la ventana de su navegador.
- Para ello, sólo es añadiremos la propiedad position: fixed.

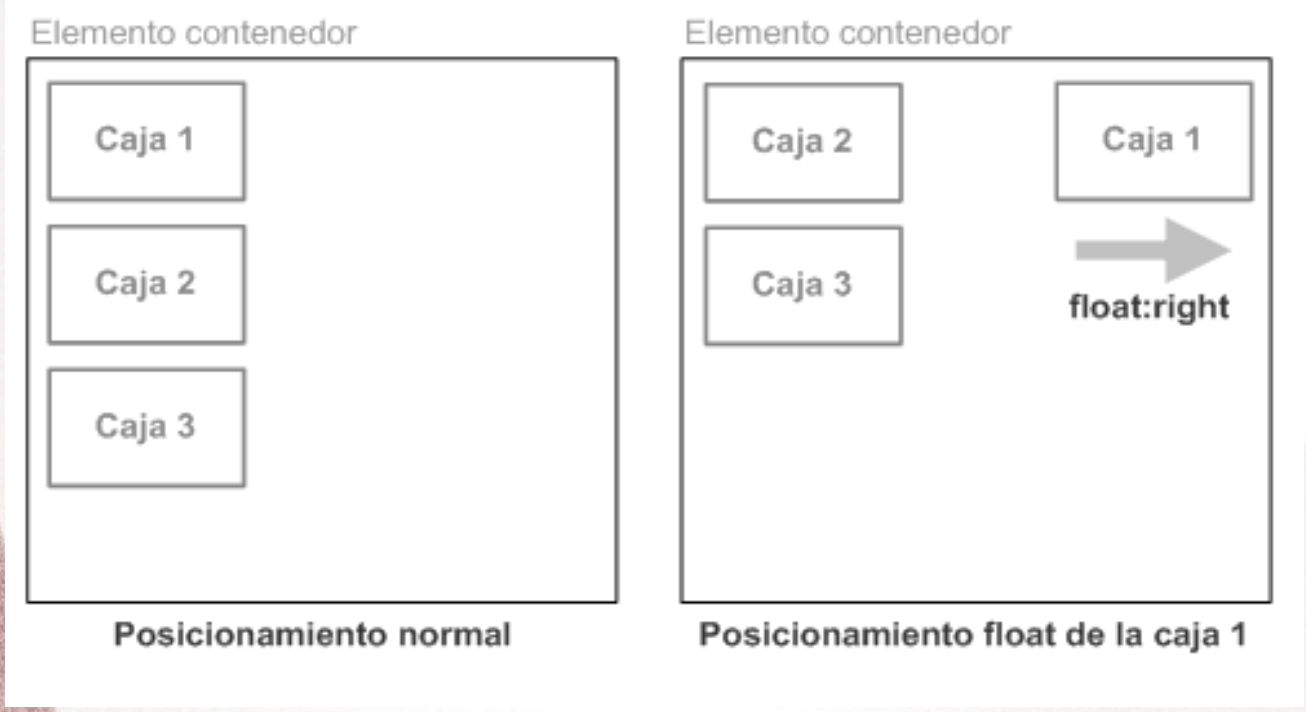
Posicionamiento flotante

- El posicionamiento flotante es el más difícil de comprender pero al mismo tiempo es el más utilizado.

- Cuando una caja se posiciona con el modelo de posicionamiento flotante se desplaza hasta la zona más a la izquierda o más a la derecha de la posición en la que originalmente se encontraba.

Posicionamiento flotante

- Ejemplo: posicionar de forma flotante hacia la derecha la caja 1:

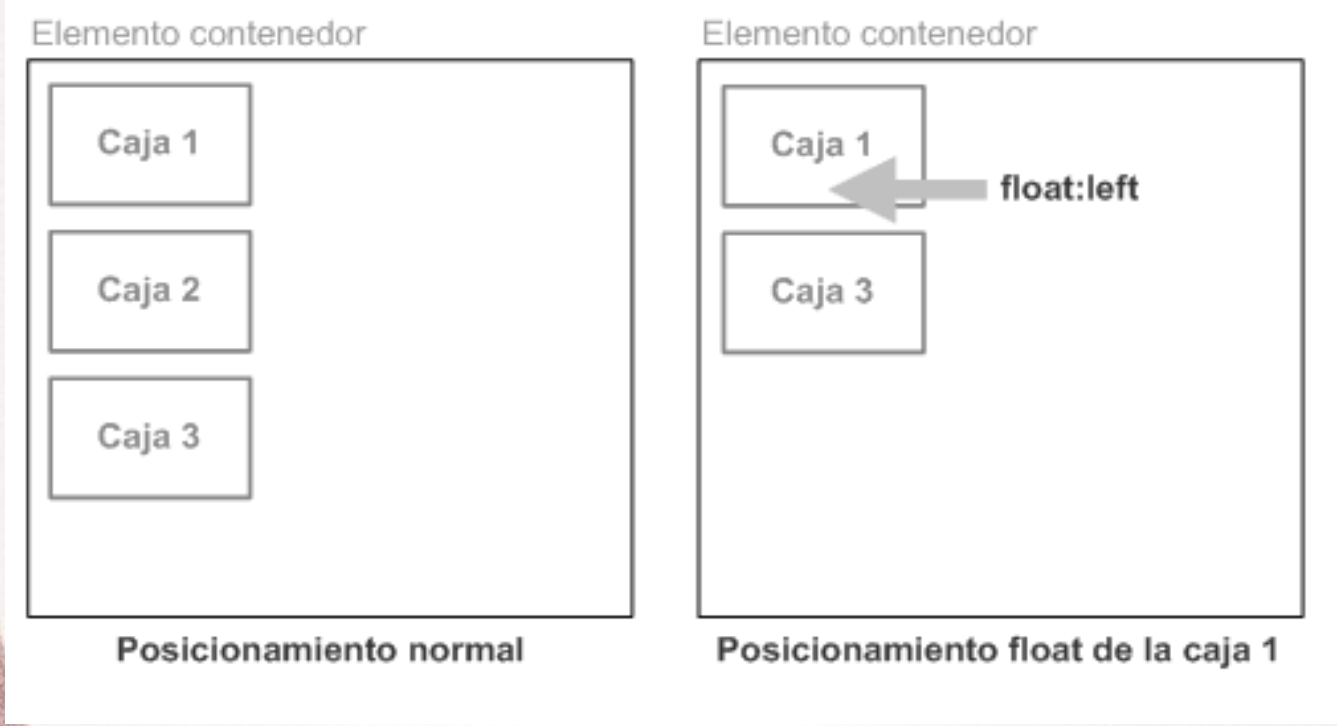


Posicionamiento flotante

- Cuando se posiciona una caja de forma flotante:
  - La caja 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 se posiciona lo más a la izquierda o lo más a la derecha posible de la posición en la que se encontraba originalmente.

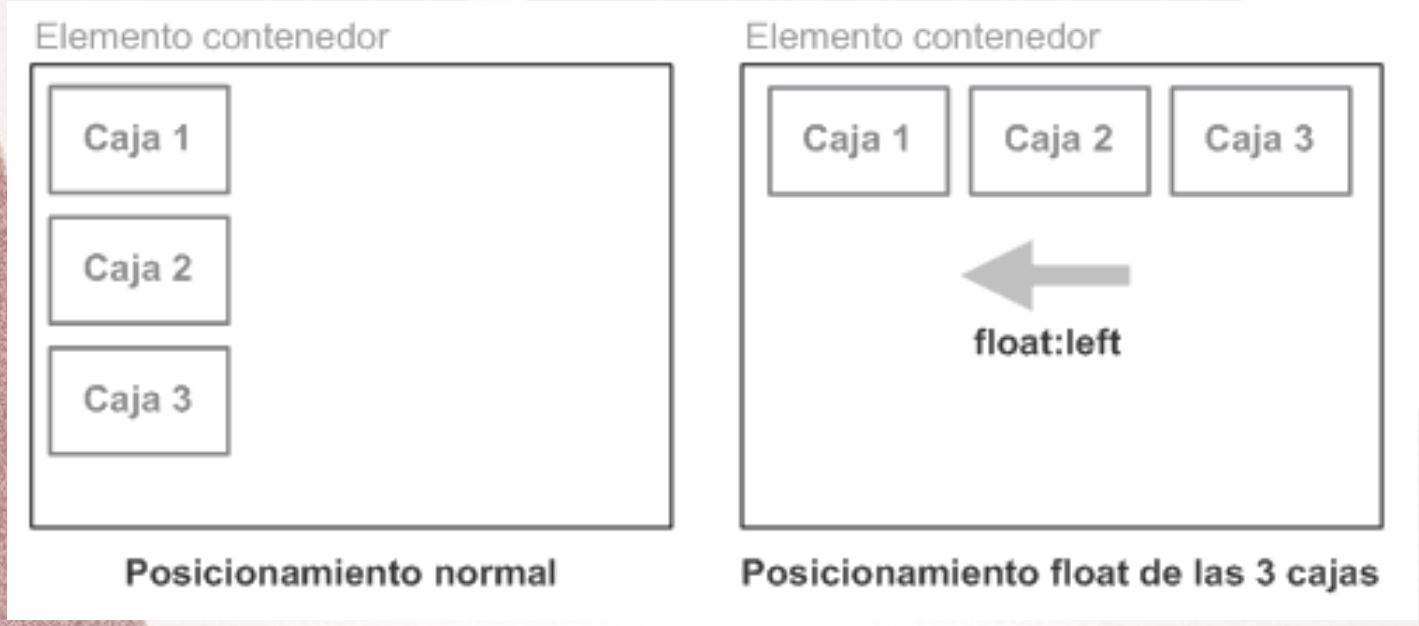
Posicionamiento flotante

- Ejemplo: la caja 1 se posiciona de forma flotante hacia la izquierda, el resultado es:



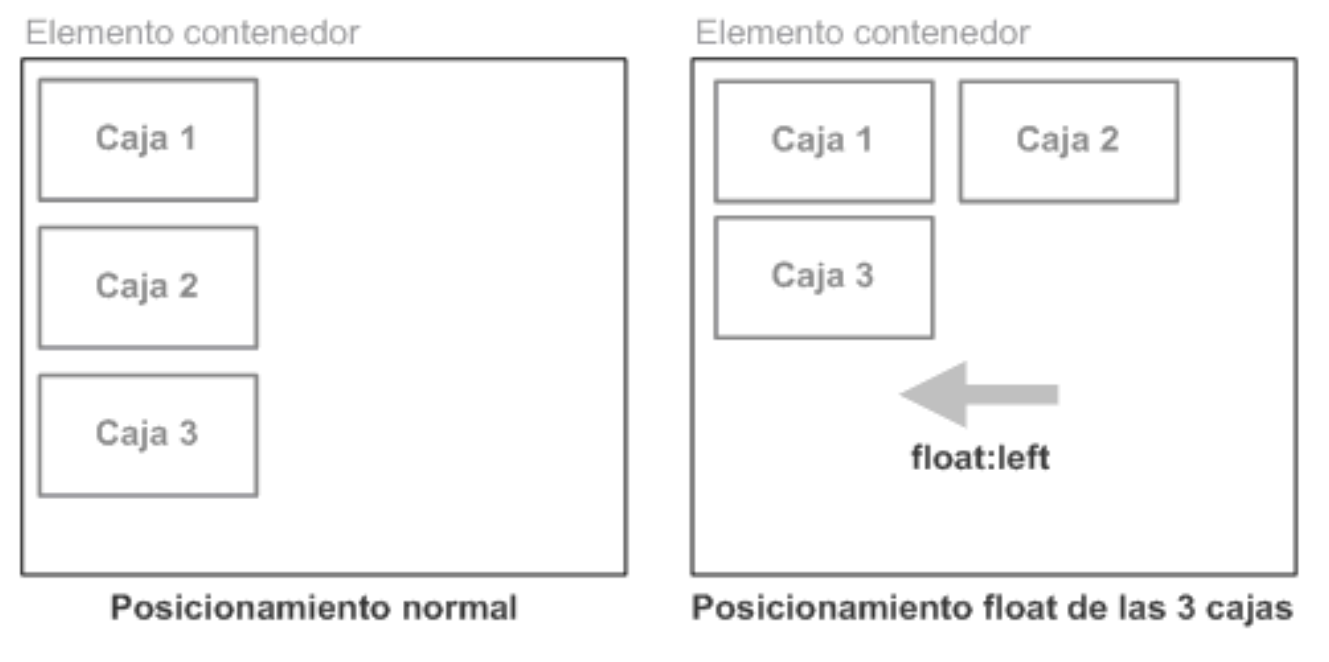
Posicionamiento flotante

- Si existen otras cajas flotantes, al posicionar de forma flotante otra caja, se tiene en cuenta el sitio disponible.



Posicionamiento flotante

- 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:



Posicionamiento flotante

- La propiedad **CSS** que permite posicionar de forma flotante una caja se denomina *float*:

`float`	Posicionamiento float
Valores	`left` | `right` | `none` | `inherit`
Se aplica a	Todos los elementos
Valor inicial	none
Descripción	Establece el tipo de posicionamiento flotante del elemento

Posicionamiento flotante

- El valor none permite anular el posicionamiento flotante de forma que el elemento se muestre en su posición original.

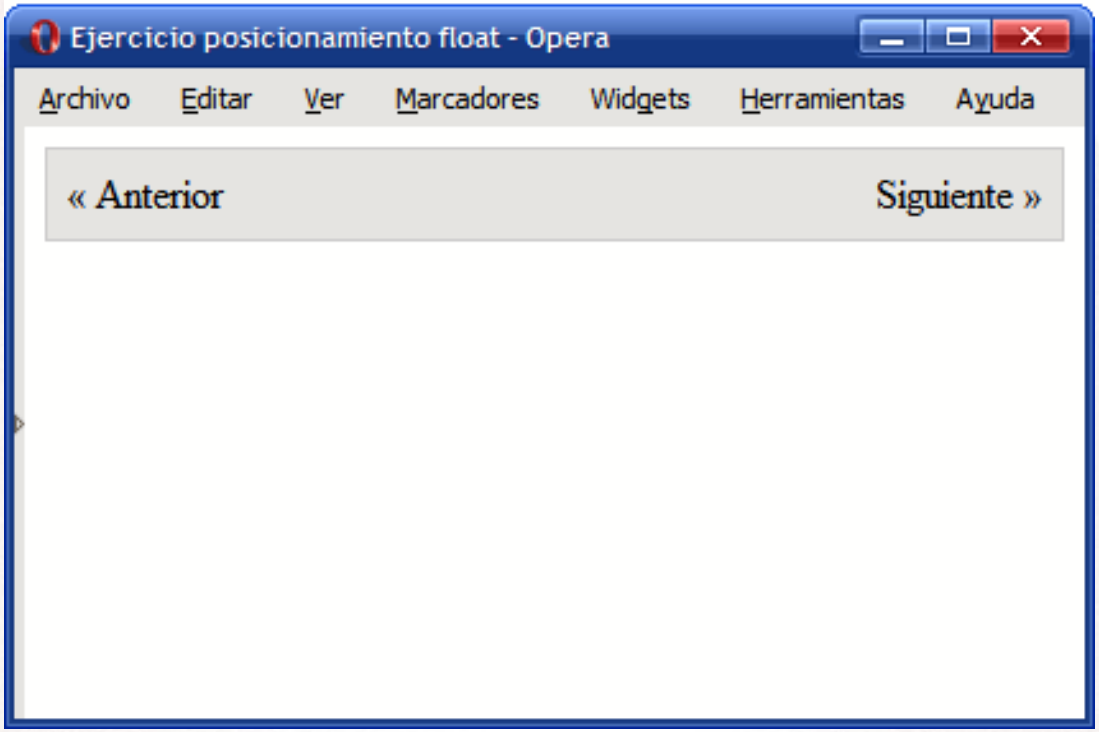
Ejercicio 6

- A partir del código HTML proporcionado:

```
<!DOCTYPE>
<html>
<head>
<title>Ejercicio posicionamiento float</title>
<style type="text/css">
</style>
</head>
<body>
<div>
    &laquo; Anterior   Siguiente &raquo;
</div>
</body>
</html>
```

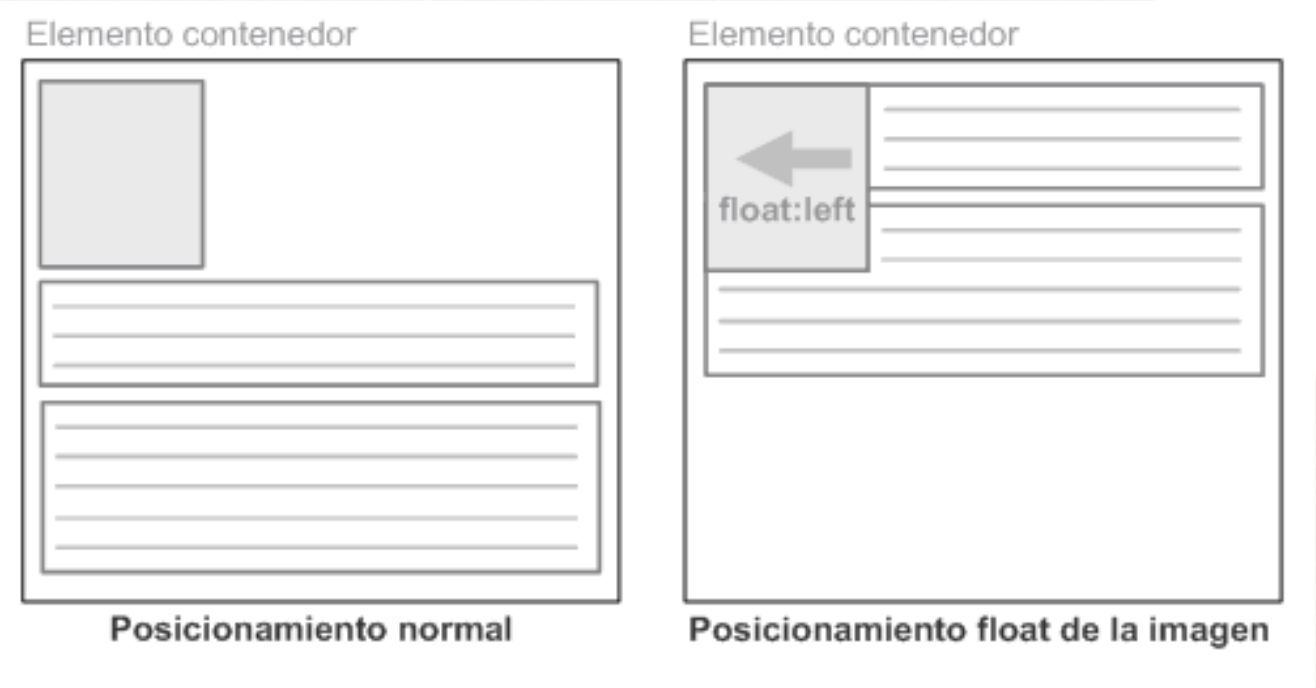
Ejercicio 6

- Determinar las reglas **CSS** necesarias para que el resultado sea similar al mostrado en la siguiente imagen:



Posicionamiento flotante

- Los elementos que se encuentran alrededor de una caja flotante adaptan sus contenidos para que fluyan alrededor del elemento posicionado:



Posicionamiento flotante

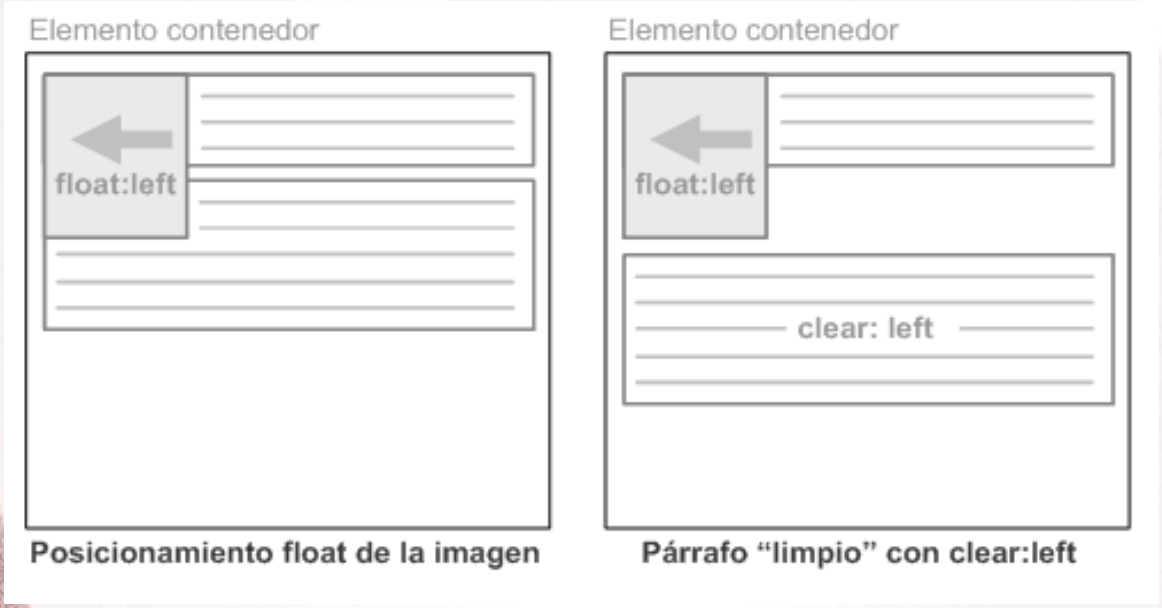
- La regla **CSS** que se aplica en la imagen del ejemplo anterior es:

```
img {
    float: left;
}
```

- 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*.

Posicionamiento flotante

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



Posicionamiento flotante

- 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:

```
<p style="clear: left;">...</p>
```

Posicionamiento flotante

- La definición formal de la propiedad **clear** se muestra a continuación:

`clear`	Despejar los elementos flotantes adyacentes
Valores	`none` | `left` | `right` | `both` | `inherit`
Se aplica a	Todos los elementos de bloque
Valor inicial	none
Descripción	Indica el lado del elemento que no debe ser adyacente a ninguna caja flotante

- Si se indica el valor **left**, el elemento se desplaza de forma descendente hasta que pueda colocarse en una línea en la que no haya ninguna caja flotante en el lado izquierdo.

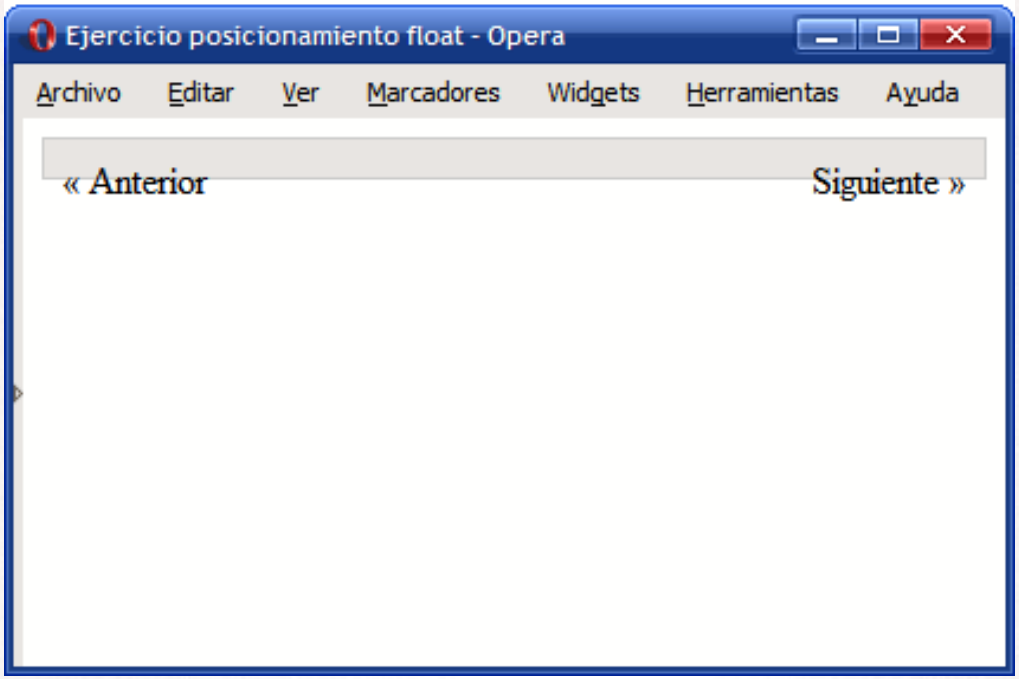
Posicionamiento flotante

- La especificación oficial de **CSS** explica este comportamiento como

"un desplazamiento descendente hasta que el borde superior del elemento esté por debajo del borde inferior de cualquier elemento flotante hacia la izquierda“.

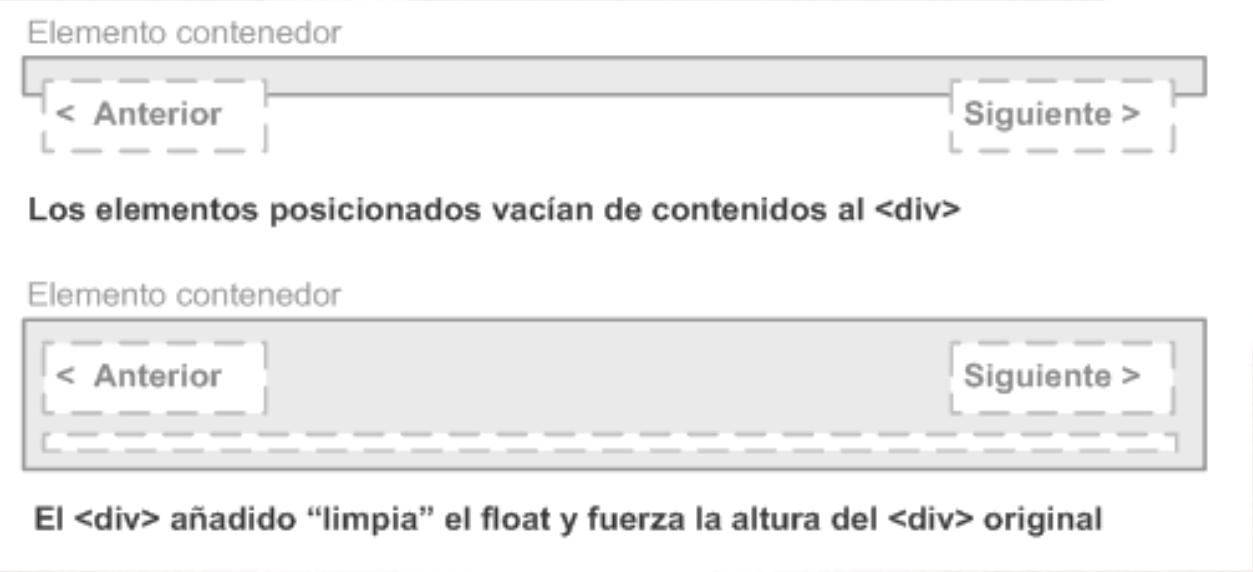
Posicionamiento flotante

- En el ejercicio anterior, se utiliza la propiedad **float** para posicionar de forma flotante los dos elementos:



Posicionamiento flotante

- Los dos elementos **** creados dentro del elemento ** ** se han posicionado mediante **float**.
- El elemento ** ** no tiene contenidos y por eso no llega a cubrir el texto de los dos elementos ****:

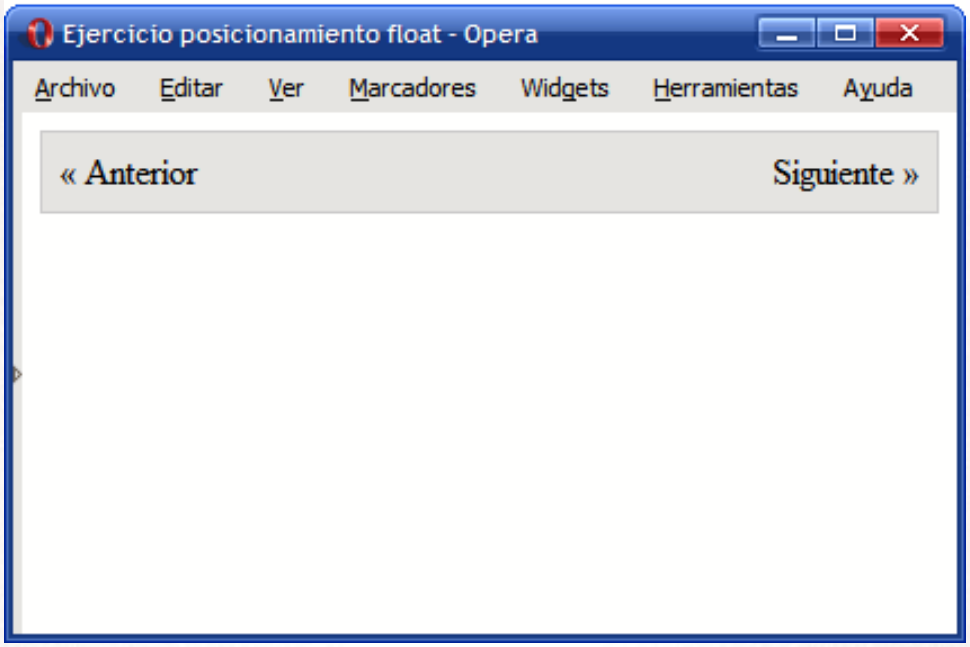


Posicionamiento flotante

- La solución consiste en añadir un elemento adicional invisible que *limpie* el **float** forzando a que el ** ** original cubra completamente los dos elementos ****.

Posicionamiento flotante

- Al añadir un ** ** con la propiedad *clear*: *both*, se tiene la seguridad de que el ** ** añadido se va a mostrar debajo de cualquier elemento posicionado con ***float***.



Posicionamiento “sticky”

- position: sticky
- Este es un valor que es nuevo relativamente para esta propiedad.
- Usando este valor, el elemento actúa como si estuviera posicionado con el valor relative (es decir, no afecta a la posición de sus elementos adyacentes) hasta que se alcanza un umbral de desplazamiento (en el propio elemento o en el elemento padre), con el cual el elemento pasa a posicionarse como si estuviera posicionado con el valor fixed.

Posicionamiento “sticky”

- Ejemplo:

```
#sidebar {
  position: -webkit-sticky;  // required for Safari
    position: sticky;
    top: 0; // required as well.
}
```

- Conforme desplacemos la página, cuando la distancia del sidebar al borde superior alcance 0, el sidebar debería quedar fijo, proporcionándonos de manera efectiva una posición fija.
- Puedes hacer pruebas en este ejemplo, poniéndole por ejemplo top: 10px para observar el efecto:

  https://codepen.io/tutsplus/pen/xGggLa

Visualización

- **CSS** define otras *cuatro propiedades* para controlar su visualización: **display**, **visibility**, **overflow** y **z-index**.

- Utilizando algunas de estas propiedades es posible *ocultar y/o hacer invisibles las cajas* de los elementos, por lo que son imprescindibles para realizar *efectos avanzados* y *animaciones*.

Propiedades display y visibility

- Las propiedades **display** y **visibility** controlan la visualización de los elementos.

- Las dos propiedades permiten *ocultar* cualquier elemento de la página.

Propiedades display y visibility

- La propiedad **display** permite *ocultar* completamente un elemento haciendo que desaparezca de la página.

- Como el elemento oculto no se muestra, *el resto de elementos* de la página *se mueven para ocupar su lugar*.

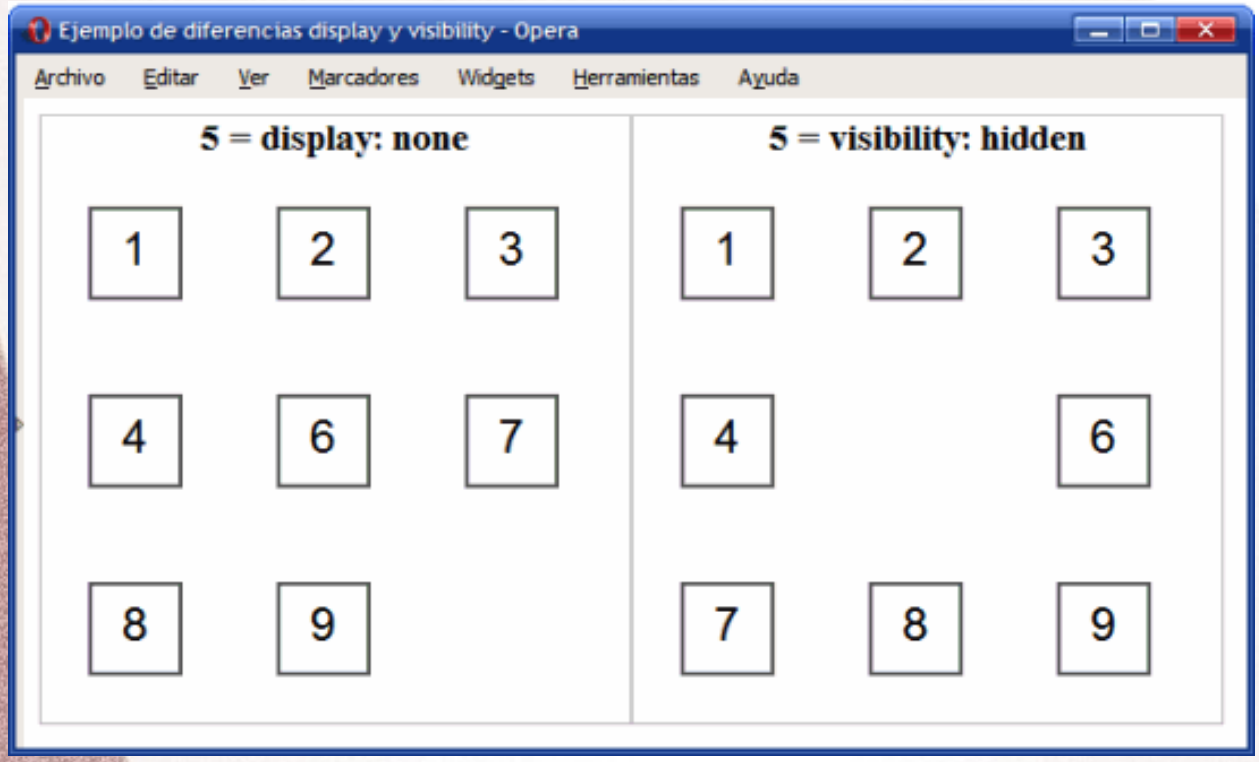
Propiedades display y visibility

- La propiedad **visibility** permite hacer *invisible* un elemento, el navegador crea la caja del elemento pero no la muestra.

- El resto de elementos de la página no modifican su posición, ya que aunque la caja no se ve, sigue ocupando sitio.

Propiedades display y visibility

- Diferencia entre la propiedad **display** y la propiedad **visibility**:



Propiedades display y visibility

- Definición de la propiedad display:

display	Visualización de un elemento
Valores	`inline | block | none | list-item | run-in | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | inherit`
Se aplica a	Todos los elementos
Valor inicial	inline
Descripción	Permite controlar la forma de visualizar un elemento e incluso ocultarlo

Propiedades display y visibility

- La propiedad **display** modifica la forma en la que se *visualiza un elemento*.

- Los valores más utilizados son **inline**, **block** y **none**.

- El valor **block** muestra un elemento como si fuera un *elemento de bloque*, independientemente del tipo de elemento que se trate.

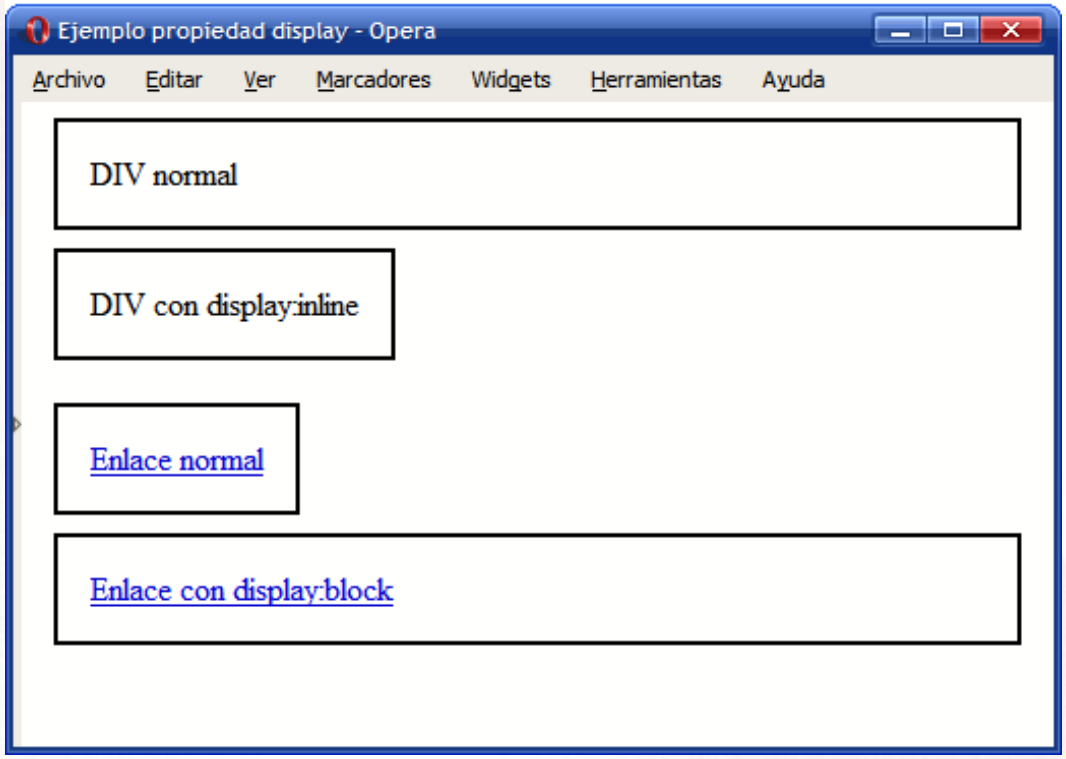
Propiedades display y visibility

- El valor **inline** visualiza un elemento en forma de *elemento en línea*.

- El valor **none** oculta un elemento y hace que *desaparezca* de la página.

Propiedades display y visibility

- Ejemplo: muestra el uso de la propiedad **display** para mostrar un elemento de bloque como si fuera un elemento en línea y a la inversa:



Propiedades display y visibility

- Las reglas **CSS** del ejemplo anterior son las siguientes:

```
<div>DIV normal</div>

<div style="display:inline">DIV con display:inline</div>

<a href="#">Enlace normal</a>

<a href="#" style="display:block">Enlace con display:block</a>
```

- La propiedad ***display: inline*** se puede utilizar en las listas (<ul>, <ol>) que se quieren mostrar *horizontalmente* y la propiedad *display: block* se emplea frecuentemente para los enlaces que forman el *menú de navegación*.

Propiedades display y visibility

- La definición completa de la propiedad **visibility** es mucho más sencilla:

`visibility`	Visibilidad de un elemento
Valores	`visible` | `hidden` | `collapse` | `inherit`
Se aplica a	Todos los elementos
Valor inicial	visible
Descripción	Permite hacer visibles e invisibles a los elementos

Propiedades display y visibility

- Empleando el valor **hidden** es posible convertir una caja en invisible para que no muestre sus contenidos.
- El valor **collapse** de la propiedad visibility sólo se puede utilizar en las filas, grupos de filas, columnas y grupos de columnas de una tabla.
  - Oculta completamente la fila y/o columna y se pueden mostrar otros contenidos en ese lugar.
- Si se utiliza el valor **collapse** sobre cualquier otro tipo de elemento, su efecto es idéntico al valor hidden.

Relación entre display, float y position

- Cuando se establecen las propiedades **display**, **float** y **position** sobre una misma caja, su interpretación es la siguiente:

1. Si display vale *none*, se ignoran las propiedades **float** y **position** y la caja no se muestra en la página.

2. Si **position** vale *absolute* o *fixed*, la caja se posiciona de forma absoluta, se considera que **float** vale *none* y la propiedad **display** vale *block* tanto para los elementos en línea como para los elementos de bloque. La *posición de la caja* se determina mediante el valor de las propiedades **top**, **right**, **bottom** y **left**.

3. En cualquier otro caso, si **float** tiene un valor *distinto* de *none*, la caja se posiciona de forma flotante y la propiedad display vale block tanto para los elementos en línea como para los elementos de bloque.

Propiedad overflow

- En algunas ocasiones el contenido de un elemento no cabe en el espacio reservado para ese elemento y se *desborda*.

- La situación más habitual: es cuando se establece la anchura y/o altura de un elemento mediante la propiedad *width* y/o *height*.

Propiedad overflow

- **CSS** define la propiedad **overflow** para controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos.

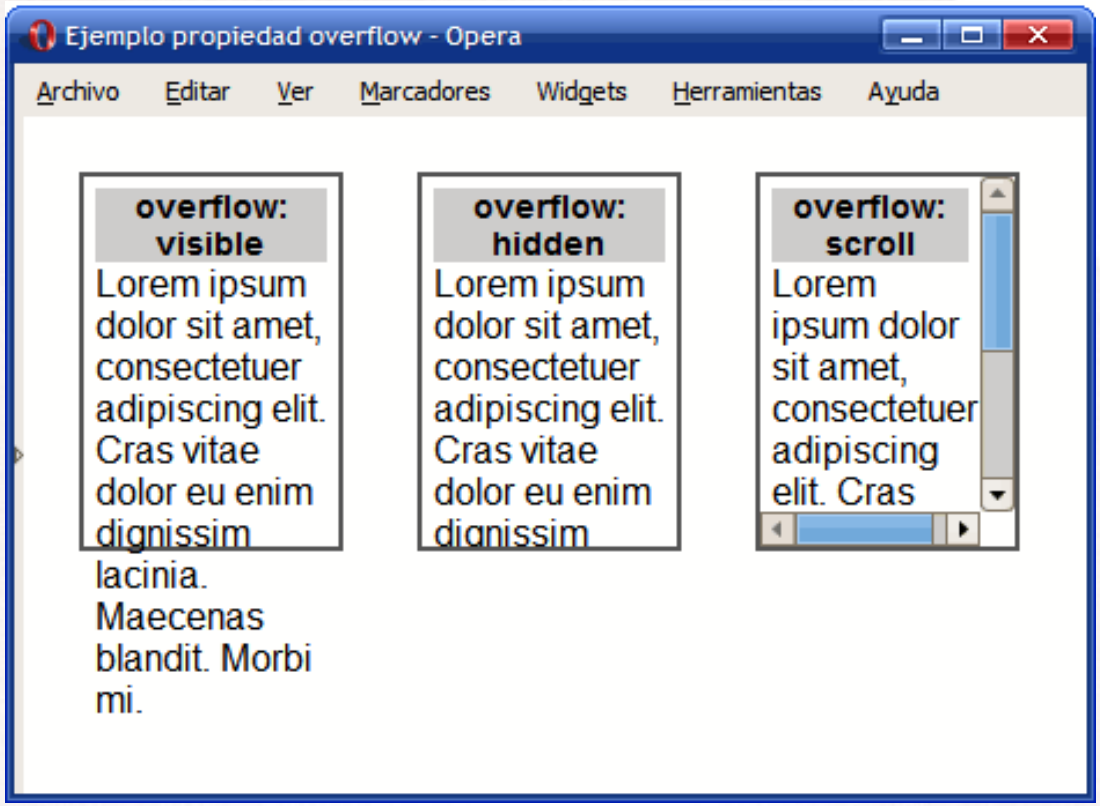
`overflow`	Parte sobrante de un elemento
Valores	`visible` | `hidden` | `scroll` | `auto` | `inherit`
Se aplica a	Elementos de bloque y celdas de tablas
Valor inicial	visible
Descripción	Permite controlar los contenidos sobrantes de un elemento

Propiedad overflow

- Los valores de la propiedad **overflow** tienen el siguiente significado:
  - **visible**: el contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento. Este es el comportamiento por defecto.
  - **hidden**: el contenido sobrante se oculta y sólo se visualiza la parte del contenido que cabe dentro de la zona reservada para el elemento.
  - **scroll**: solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero también se muestran barras de *scroll* que permiten visualizar el resto del contenido.
  - **auto**: el comportamiento depende del navegador, aunque normalmente es el mismo que la propiedad scroll.

Propiedad overflow

- La siguiente imagen muestra un ejemplo de los tres valores típicos de la propiedad **overflow:**



Propiedad overflow

- El código **HTML** y **CSS** del ejemplo anterior se muestran a continuación:

```
div {
    display: inline;
    float: left;
    margin: 1em;
    padding: .3em;
    border: 2px solid #555;
    width: 100px;
    height: 150px;
    font: 1em Arial, Helvetica, sans-serif;
}
```

Propiedad overflow

<h1>overflow: visible</h1> Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras vitae dolor eu enim dignissim lacinia. Maecenas blandit. Morbi mi.

<h1>overflow: hidden</h1> Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras vitae dolor eu enim dignissim lacinia. Maecenas blandit. Morbi mi.

<h1>overflow: scroll</h1> Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Cras vitae dolor eu enim dignissim lacinia. Maecenas blandit. Morbi mi.

Propiedad z-index

- **CSS** permite controlar la posición *tridimensional* de las cajas posicionadas.

- Es posible indicar las cajas que 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.

Propiedad z-index

- A continuación se muestra la definición formal de la propiedad **z-index**:

z-index	Orden tridimensional
Valores	auto | <numero> | inherit
Se aplica a	Elementos que han sido posicionados explícitamente
Valor inicial	auto
Descripción	Establece el nivel tridimensional en el que se muestra el elemento

- El valor más común de la propiedad **z-index** es un número entero.

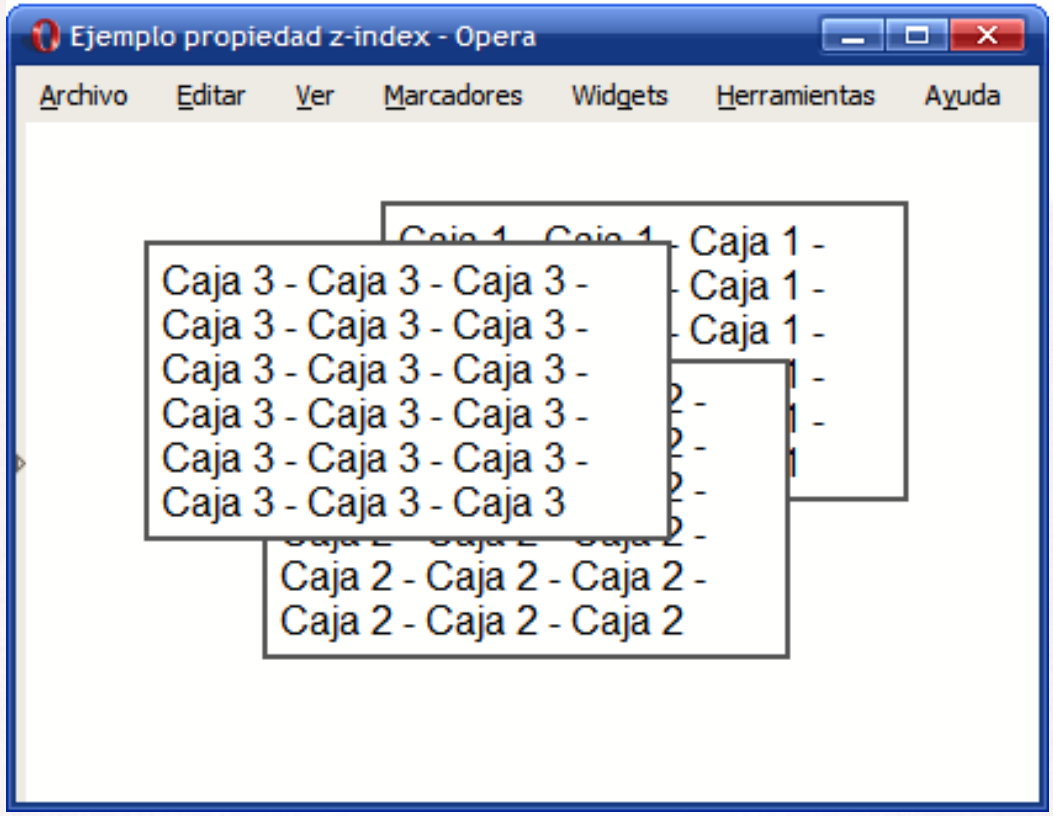
Propiedad z-index

- Cuanto **más alto** sea el valor numérico, *más cerca del usuario* se muestra la caja.

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

Propiedad z-index

- La siguiente imagen muestra un ejemplo de uso de la propiedad **z-index**:



Propiedad z-index

- El código **HTML** y **CSS** del ejemplo anterior es el siguiente:

```
div { position: absolute; }
#caja1 { z-index: 5; top: 1em; left: 8em;}
#caja2 { z-index: 15; top: 5em; left: 5em;}
#caja3 { z-index: 25; top: 2em; left: 2em;}

<div id="caja1">Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 -
Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 - Caja 1 -
Caja 1 - Caja 1 - Caja 1 - Caja 1</div>
<div id="caja2">Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 -
Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 - Caja 2 -
Caja 2 - Caja 2 - Caja 2 - Caja 2</div>
<div id="caja3">Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 –
Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 - Caja 3 -
Caja 3 - Caja 3 - Caja 3 - Caja 3</div>
```

Propiedad z-index

- La propiedad **z-index** sólo tiene efecto en los elementos posicionados, por lo que es obligatorio que la propiedad **z-index** vaya acompañada de la propiedad **position**.

- Si debes posicionar un elemento pero no quieres moverlo de su posición original ni afectar al resto de elementos de la página, puedes utilizar el posicionamiento relativo (***position: relative***).