

Programación Estática y Laboratorio web

CSS

Parte III



Universidad Nacional del Comahue
Facultad de Economía y Administración
Departamento de Ciencias de la Computación

[Modelo de Cajas]

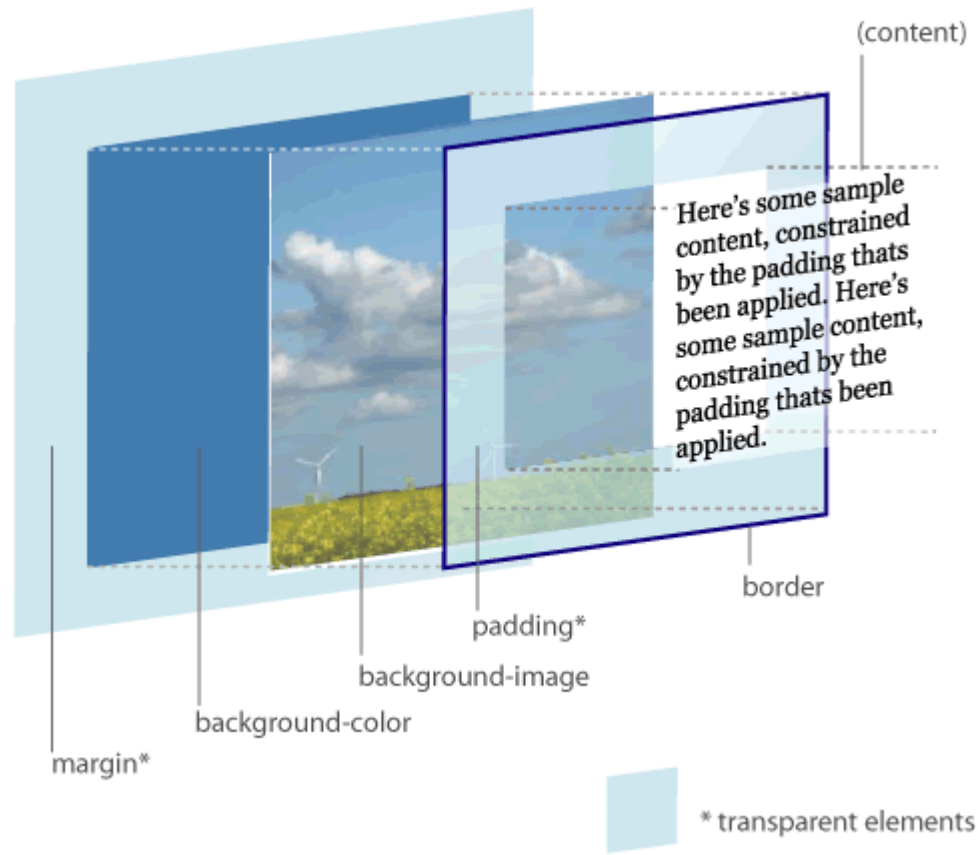
- Repasamos un poco <http://www.csszengarden.com/>
- Vemos que si bien las cosas cambian de color, tipo de letra, fondo e incluso de ubicación, siempre se mueven en “bloques”.
- “Modelo de cajas”.

[Modelo de Cajas]

- El “box model” es una de las características más importantes del CSS.
- Mediante html todas las cosas se representan mediante cajas rectangulares.
- Mediante CSS se controlan las propiedades de las cajas y también su representación visual (width, height, border).
- CSS permite además definir el margen que se deja entre cada caja y el espacio de relleno interior que muestra cada caja.
- También permite controlar controlar la forma en la que se visualizan las cajas: se pueden ocultar, desplazar respecto de su posición original, fijarlas en una posición concreta dentro del documento, etc.

[Modelo de Cajas]

THE CSS BOX MODEL HIERARCHY



[Modelo de Cajas]

- **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): está formado por el espacio libre entre el contenido y el borde que lo encierra.
 - Borde (border): línea que encierra completamente el contenido y su relleno.
 - Imagen de fondo (background image): imagen que se muestra por debajo del contenido. Si se define un color y una imagen de fondo, la imagen tiene más prioridad y es la que se visualiza.
 - Color de fondo (background color): color que rellena el espacio ocupado por el contenido y su posible relleno. Si se define un color y una imagen de fondo, el color tiene menos prioridad y por tanto se visualiza la imagen.
 - Margen (margin): espacio libre entre la caja y las posibles cajas adyacentes.

[Modelo de Cajas]

- El relleno y el margen son transparentes, por lo que en el espacio ocupado por el relleno se muestra el color o imagen de fondo (si están definidos) y en el espacio ocupado por el margen se muestra el color o imagen de fondo de su elemento padre (si están definidos). Si ningún elemento padre tiene definido un color o imagen de fondo, se mostraría el color o imagen de fondo de la propia página (si están definidos).

[Volvemos a HTML: div]

- Para agrupar los elementos que forman cada zona de la página se utiliza `<div>`.
- Es una opción superadora a las tablas en lo que se refiere a estructurar una página.
- `div` = división.
- Sin embargo, se lo suele referenciar como “capa” o “layer”.
- Generalmente las páginas complejas tienen decenas de `<div>`, referenciadas mediante `id` o `class`.

[Etiqueta div]

■ Ejemplo:

```
<div>



<br/>


</div>
```


[Etiqueta span]

- Permite aplicar estilos de manera inline dentro de contenido html.

This is `classified` text

Estilos. fondos: background-color

| | |
|-------------------------|--|
| background-color | Color de fondo |
| Valores | <color> transparent inherit |
| Se aplica a | Todos los elementos |
| Valor inicial | transparent |
| Descripción | Establece un color de fondo para los elementos |

■ Ejemplos:

```
body { background-color: #F5F5F5; }
```

Estilos. fondos: background-image

| | |
|-------------------------|--|
| background-image | Imagen de fondo |
| Valores | <url> none inherit |
| Se aplica a | Todos los elementos |
| Valor inicial | none |
| Descripción | Establece una imagen como fondo para los elementos |

- CSS permite establecer de forma simultánea un color y una imagen de fondo. En este caso, la imagen se muestra delante del color, por lo que solamente si la imagen contiene zonas transparentes es posible ver el color de fondo.

■ Ejemplos:

```
body { background-image: url("imagenes/fondo.png") }
```

Estilos. fondos: background-repeat

| | |
|--------------------------|--|
| background-repeat | Repetición de la imagen de fondo |
| Valores | repeat repeat-x repeat-y no-repeat inherit |
| Se aplica a | Todos los elementos |
| Valor inicial | repeat |
| Descripción | Controla la forma en la que se repiten las imágenes de fondo |

■ Ejemplos:

```
#hdr {  
    background-image: url(images/fondo.gif);  
    background-repeat: repeat-x;  
    width: 100%;  
    text-align: center;  
}
```

Estilos. fondos: background-position

| | |
|----------------------------|---|
| background-position | Posición de la imagen de fondo |
| Valores | ((<porcentaje> <medida> left center right) (<porcentaje> <medida> top center bottom)?) ((left center right) (top center bottom)) inherit |
| Se aplica a | Todos los elementos |
| Valor inicial | 0% 0% |
| Descripción | Controla la posición en la que se muestra la imagen en el fondo del elemento |

- Permite indicar qué distancia se desplaza la imagen de fondo respecto de su posición original situada en la esquina superior izquierda.
- Si se indican dos porcentajes o dos medidas, el primero indica el desplazamiento horizontal y el segundo el desplazamiento vertical respecto del origen

Estilos. fondos: background-position

```
#caja1 {  
  background-image: url(images/help.png);  
  background-repeat: no-repeat;  
  background-position: bottom left;  
}
```

```
#caja2 {  
  background-image: url(images/help.png);  
  background-repeat: no-repeat;  
  background-position: right top;  
}
```

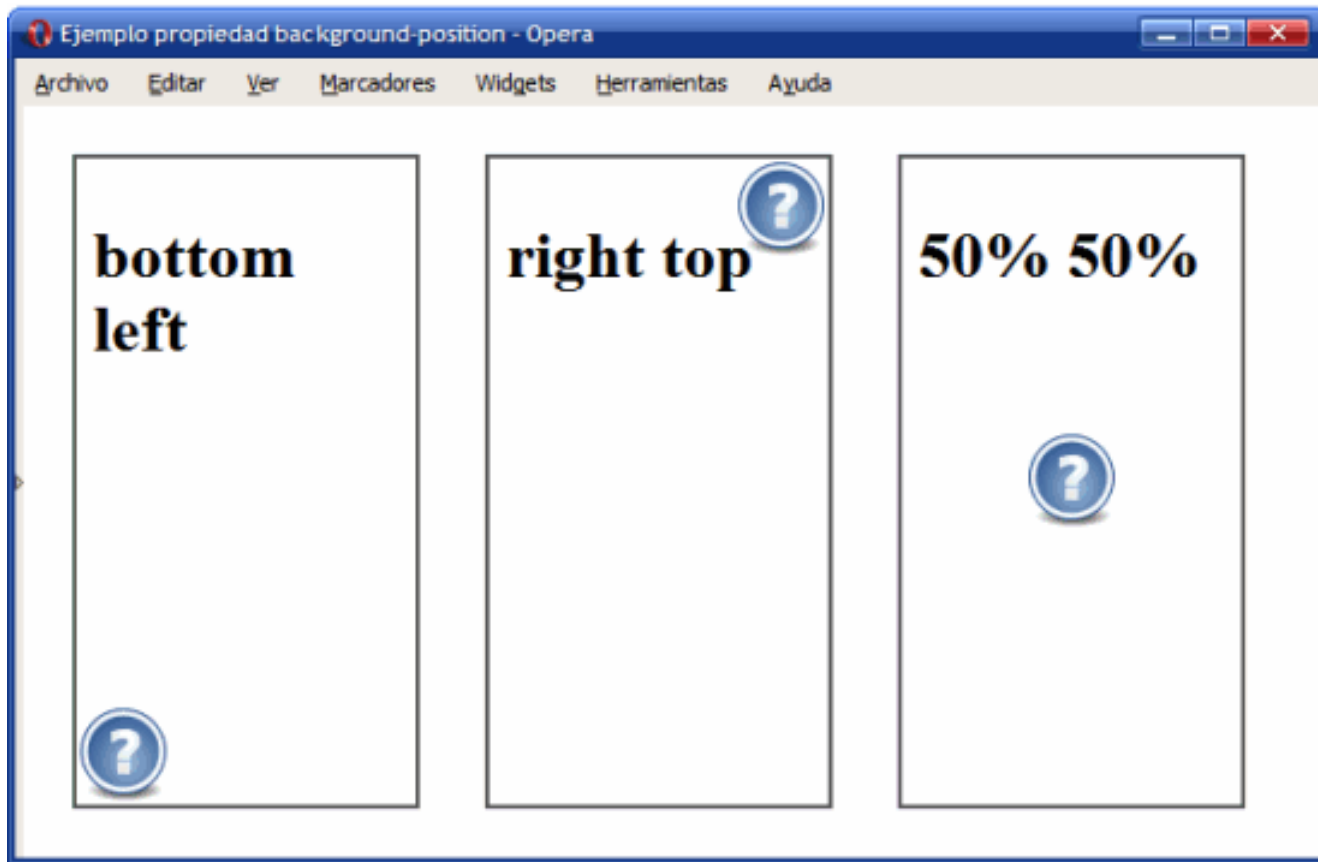
```
#caja3 {  
  background-image: url(images/help.png);  
  background-repeat: no-repeat;  
  background-position: 50% 50%;  
}
```

```
<div id="caja1"><h1>bottom left</h1></div>
```

```
<div id="caja2"><h1>right top</h1></div>
```

```
<div id="caja3"><h1>50% 50%</h1></div>
```

Estilos. fondos: background-position



[Estilos. fondos: background]

| | |
|----------------------|---|
| background | Fondo de un elemento |
| Valores | (<background-color> <background-image> <background-repeat> <background-attachment> <background-position>) inherit |
| Se aplica a | Todos los elementos |
| Valor inicial | - |
| Descripción | Establece todas las propiedades del fondo de un elemento |

- El orden en el que se indican las propiedades es indiferente, aunque en general se sigue el formato indicado de color, url de imagen, repetición y posición.

[Estilos. fondos: background]

- Los siguientes estilos son equivalentes:

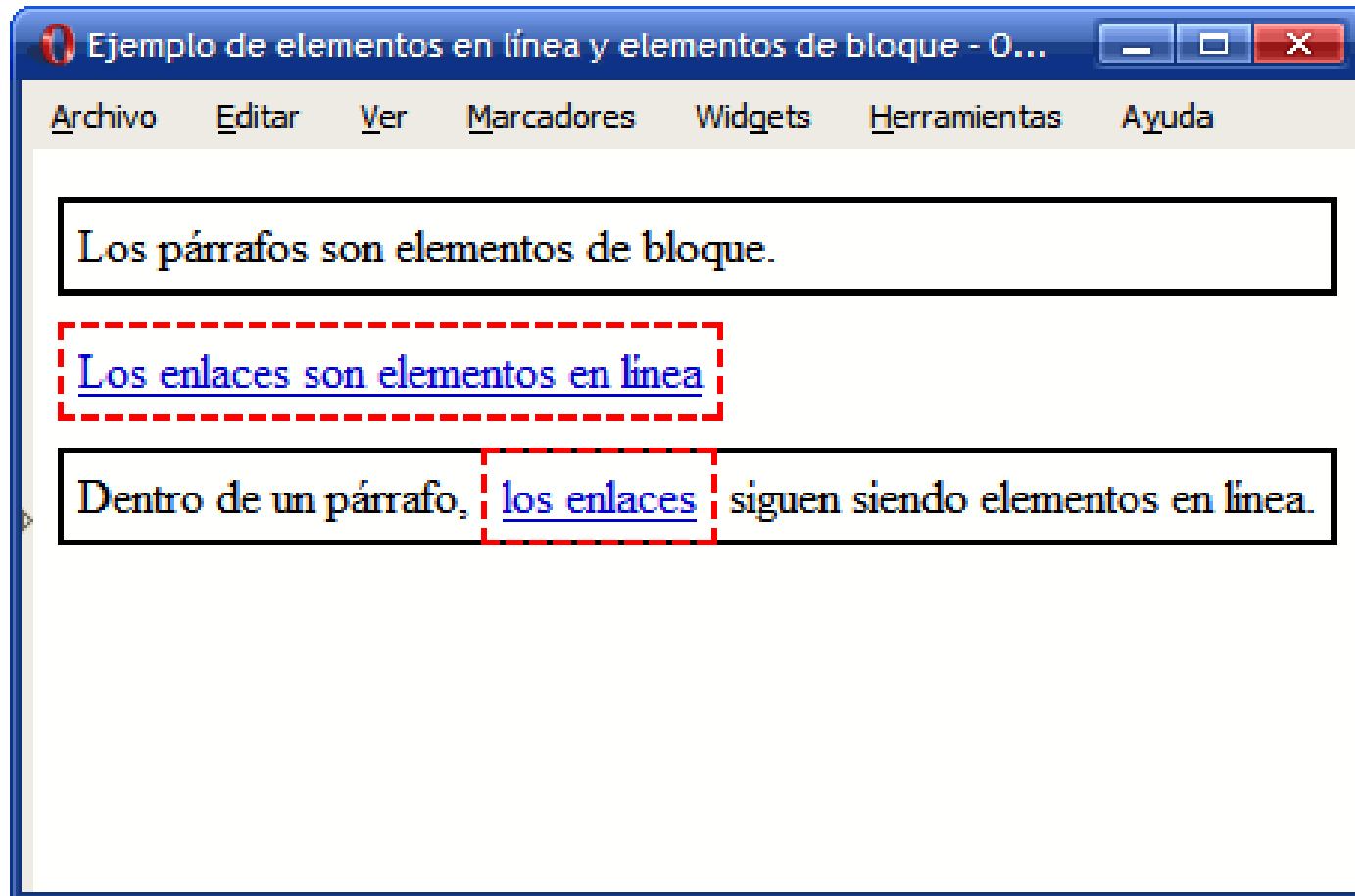
```
/* Color e imagen de fondo de la página mediante una propiedad shorthand */  
body { background: #222d2d url(/graphics/colorstrip.gif) repeat-x 0 0; }
```

```
/* La propiedad shorthand anterior es equivalente a las siguientes propiedades */  
body {  
    background-color: #222d2d;  
    background-image: url(/graphics/colorstrip.gif);  
    background-repeat: repeat-x;  
    background-position: 0 0;  
}
```

Estilos. Posicionamiento y Visualización

- Los elementos HTML se clasifican 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 sin embargo no empiezan en nueva línea y sólo ocupan el espacio ocupado por sus contenidos.
- Un elemento de bloque no puede aparecer dentro de un elemento en línea. En cambio, un elemento en línea puede aparecer dentro de un elemento de bloque y dentro de otro elemento en línea.

Estilos. Posicionamiento y Visualización



Estilos. Posicionamiento y Visualización

- Cuando los navegadores cargan una página web, muestran sus elementos de la mejor manera posible teniendo en cuenta las limitaciones de espacio y el tipo de cada elemento (de bloque o en línea).
- Además de esta forma automática, CSS permite controlar de forma precisa el lugar en el que se coloca cada elemento

Estilos. Posicionamiento y Visualización

- CSS define tres mecanismos para posicionar las cajas de cada elemento HTML:
- Normal: posicionamiento por defecto de los elementos en línea y los de bloque. Además, incluye el posicionamiento relativo.
- Float: posicionamiento que consiste en posicionar el elemento según el esquema normal y una vez colocado, desplazarlo todo lo posible hacia la izquierda o hacia la derecha.
- Absoluto: posicionamiento que consiste en extraer por completo el elemento de su posicionamiento normal y colocarlo en la posición indicada respecto de su elemento padre posicionado.

[Estilos. posicionamiento: position]

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

[Estilos. posicionamiento: position]

- static: es el posicionamiento que se utiliza por defecto y todos los elementos inicialmente se muestran de esta forma. No se tienen en cuenta los valores de las propiedades top, right, bottom y left.
- relative: la nueva posición del elemento se calcula a partir de la posición que tendría si no se utilizara la propiedad position. Las posiciones de los elementos contiguos no se ven afectadas por el desplazamiento de este elemento.
- absolute: la nueva posición del elemento se determina mediante las propiedades top, right, bottom y left. Los valores de esas propiedades indican la posición del elemento respecto de la posición de su elemento padre. La posición del siguiente elemento se calcula como si no existiera el elemento que se desplaza, ya que este último se desentiende por completo del posicionamiento normal.

[Estilos. posicionamiento: position]

- fixed: la nueva posición del elemento se calcula de forma idéntica al posicionamiento absoluto. La diferencia reside en que en el caso de fixed, el elemento no se mueve cuando se desplaza la ventana del navegador. En los medios visuales (como la pantalla) el elemento se muestra en una posición fija independiente del movimiento de la ventana del navegador. En los medios impresos, el elemento se muestra en todas las páginas.
- sticky: Es tratado como un elemento posicionado relativamente hasta que su bloque contenedor cruza un límite establecido, a partir del cual es tratado como fixed.

[Estilos. posicionamiento]

| | |
|----------------------|---|
| top | Desplazamiento superior |
| right | Desplazamiento lateral derecho |
| bottom | Desplazamiento inferior |
| left | 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 |

[Estilos. posicionamiento:]

- En el caso del posicionamiento relativo, cada una de estas propiedades indica el desplazamiento del elemento desde su borde superior/derecho/inferior/izquierdo. Si el posicionamiento es absoluto, las propiedades indican el desplazamiento del elemento respecto del borde superior/derecho/inferior/izquierdo de su elemento padre.
- En cualquiera de los dos casos, si el desplazamiento se indica en forma de porcentaje, se refiere al porcentaje sobre la anchura (propiedades right y left) o altura (propiedades top y bottom) del elemento.

Posicionamiento normal

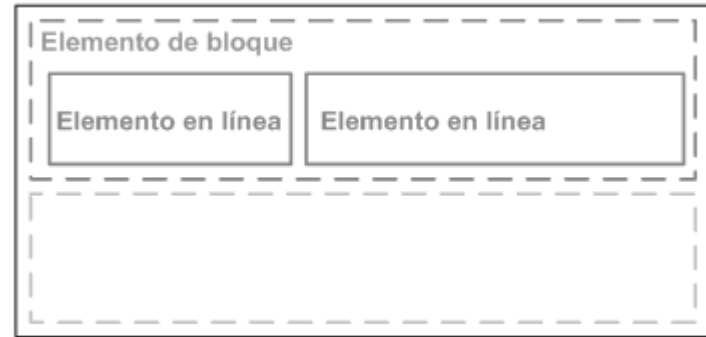
- El **posicionamiento normal** es la estrategia que utilizan por defecto los navegadores para mostrar los elementos de las páginas.
- Hasta ahora así es como hemos estado acomodando las páginas.
- Los elementos de bloque se muestran uno debajo de otro comenzando desde el principio del elemento padre. La distancia de las cajas se controla mediante los márgenes verticales.
- Los elementos de línea muestran uno detrás de otro de forma horizontal comenzando desde la posición más a la izquierda del elemento padre. La distancia entre las cajas se controla mediante los márgenes laterales.

Posicionamiento normal



Elemento contenedor

**Posicionamiento normal
de los elementos de
bloque**

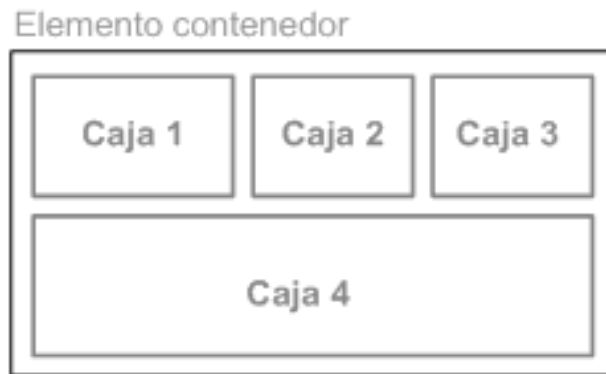


Elemento contenedor

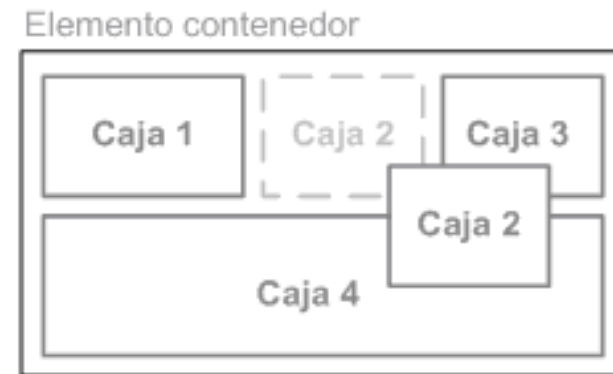
**Posicionamiento normal
de los elementos de línea**

Posicionamiento relativo

- Relativo: se considera parte del posicionamiento normal.
- Consiste en desplazar un elemento respecto de su posición original.
- El desplazamiento de una caja mediante desplazamiento relativo no afecta al resto.



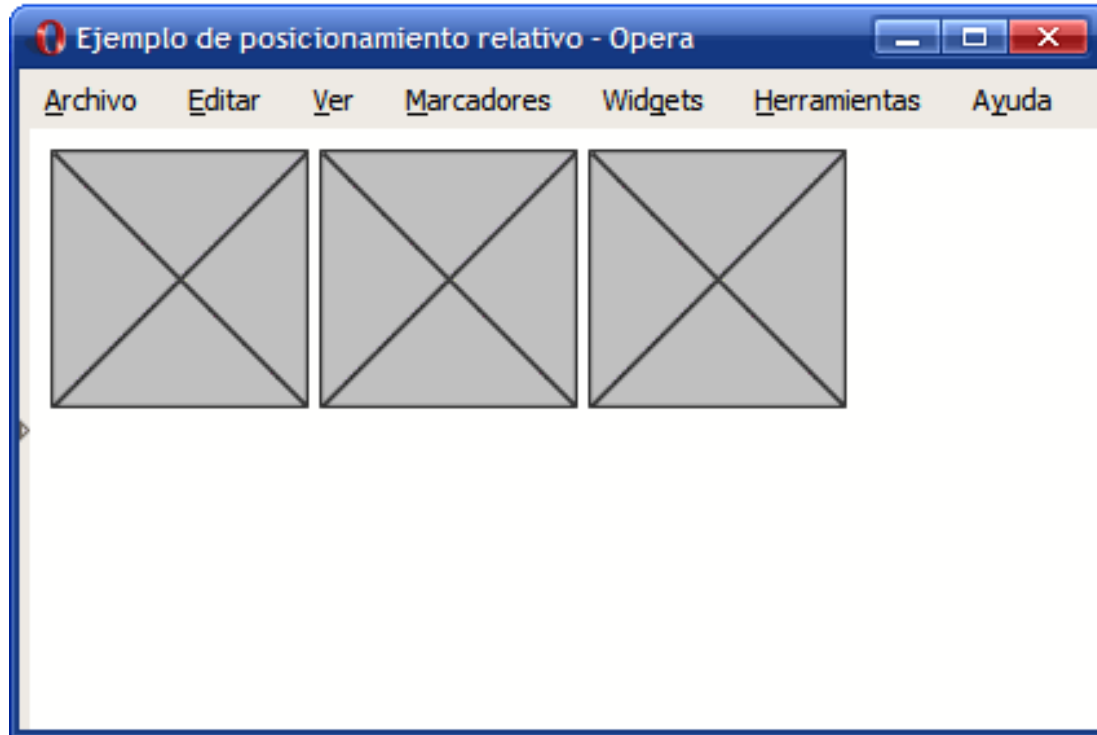
Posicionamiento normal



Posicionamiento relativo de la caja 2

[Posicionamiento relativo]

- Ejemplo:



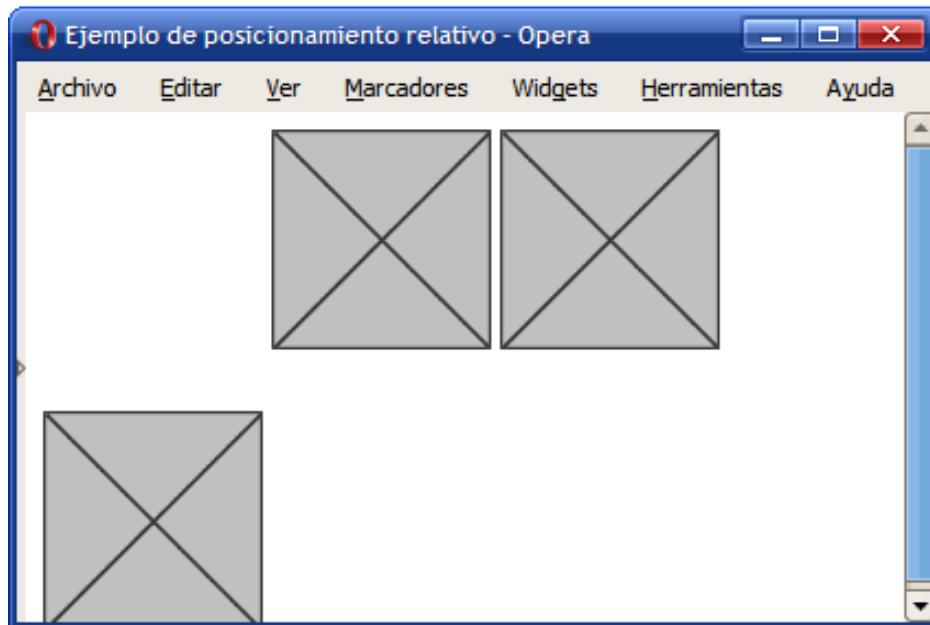
Posicionamiento relativo

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

``

``

``



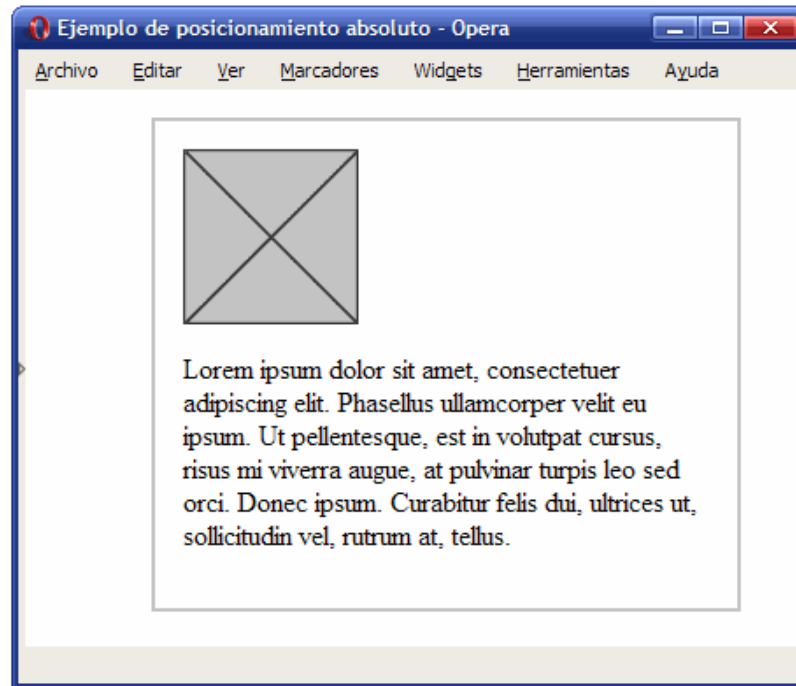
[Posicionamiento absoluto]

- Implica dar un valor de referencia x,y (top y left) donde se posicionará el elemento.
- Se toma como referencia el primer elemento padre posicionado (podría ser HTML)
- Implica que el elemento desplazado sale por completo del flujo normal del documento y por tanto, la posición del resto de elementos se determina como si no existiera el elemento desplazado.

Posicionamiento absoluto

- Implica dar un valor de referencia x,y (top y left) donde se posicionará el elemento.
- Se toma como referencia el primer elemento padre **posicionado**. Si ningún elemento padre tiene un posicionamiento se toma el documento HTML completo.
- Implica que el elemento desplazado sale por completo del flujo normal del documento y por tanto, la posición del resto de elementos se determina como si no existiera el elemento desplazado.

Posicionamiento absoluto



Ejemplo: Situación original

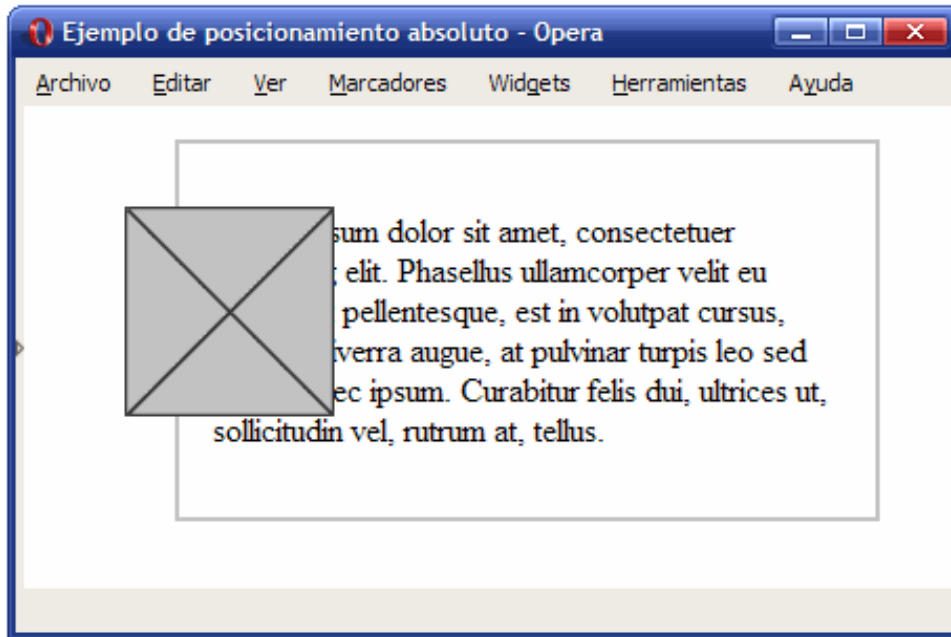
Posicionamiento absoluto

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

```
<div>

<p>Lorem ipsum dolor sit amet, consectetur 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>
```

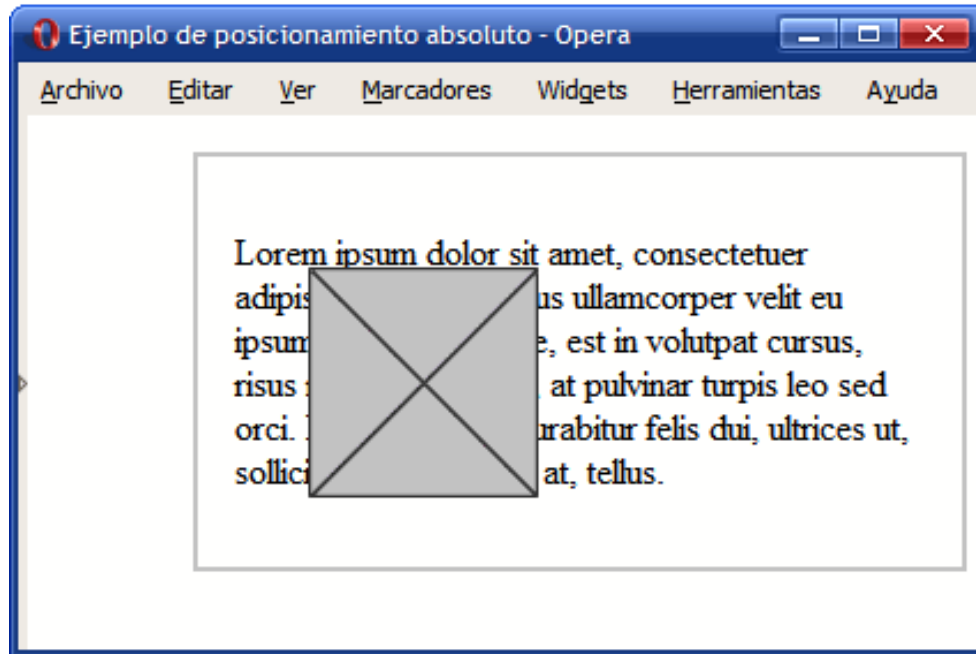
Posicionamiento absoluto



```
div img {  
  position: absolute;  
  top: 3em;  
  left: 3em;  
}
```

- Agregando el css la img se posición de forma absoluta con respecto al elemento html.
- Si div huvbiera tenido un posicionamiento, se hubiera posicionado con respecto a div.

Posicionamiento absoluto



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

- Al estar ahora posicionado el elemento padre, la imagen se posiciona referenciado a este (div)

[Posicionamiento fixed]

- Caso especial de posicionamiento absoluto.
- Los elementos que utilizan un posicionamiento fijo se muestran de forma estática en la ventana del navegador y no varían su posición ni aunque se desplace la ventana del navegador o se haga *scroll* sobre ella.

[Posicionamiento sticky]

- Funciona como relativo hasta que ocurre un scroll y el elemento padre sale de ciertos límites.
- A partir de ese momento se comporta como fixed.

[Posicionamiento]

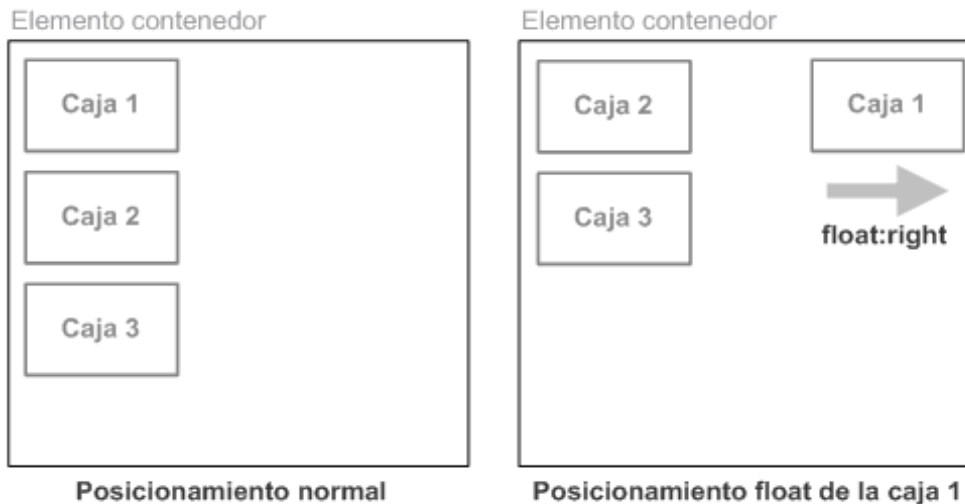
- <https://developer.mozilla.org/es/docs/Web/CSS/position>

Posicionamiento float

- Una caja posicionada mediante la propiedad float, se desplaza hasta la zona más a la izquierda o más a la derecha de la línea en la que se debería mostrar si no se desplazara.
- Una caja desplazada mediante float no pertenece al posicionamiento normal de un documento, por lo que los elementos de bloque anteriores y posteriores se visualizan como si la caja desplazada no existiera.

Posicionamiento float

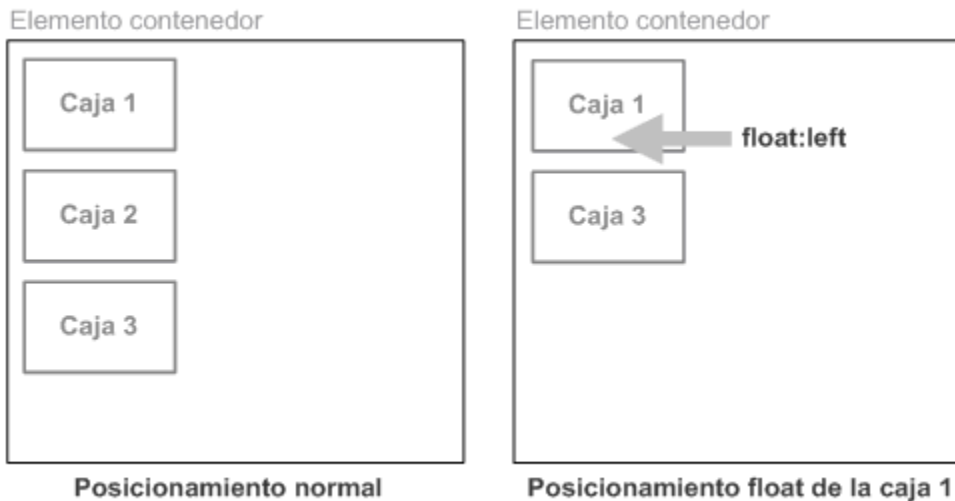
■ Ejemplo 1



- Caja 1 posicionada a la derecha. Además, la caja uno es ignorada para el posicionamiento normal.

Posicionamiento float

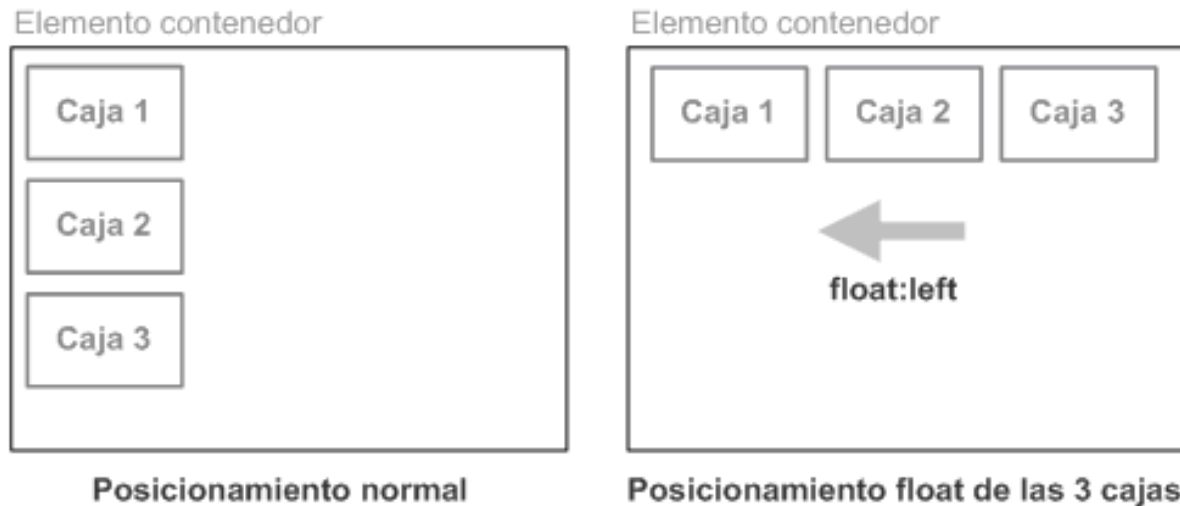
■ Ejemplo 2



- Caja 1 posicionada a la izquierda. Puede pasar que la caja 2 quede oculta

Posicionamiento float

■ Ejemplo 3



- Las tres cajas posicionadas a la izquierda.

Estilos. Posicionamiento: float

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

- Uno de los principales motivos para la creación del posicionamiento float fue 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.

[Posicionamiento float]

- <https://developer.mozilla.org/en-US/docs/Web/CSS/float>

Estilos. Posicionamiento: clear

| | |
|----------------------|---|
| clear | Despejar los elementos adyacentes |
| Valores | none left right both inherit |
| Se aplica a | Todos los elementos de bloque |
| Valor inicial | none |
| Descripción | Permite limpiar el lado derecho, izquierdo o ambos de un elemento, para que no se muestren elementos adyacentes |

- La propiedad clear indica el lado del elemento HTML que no debe ser adyacente a otros elementos desplazados mediante float.
- 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 ningún elemento en el lado izquierdo.
- *“Un desplazamiento descendente hasta que el borde superior del elemento esté por debajo del borde inferior de cualquier elemento desplazado hacia la izquierda”.*

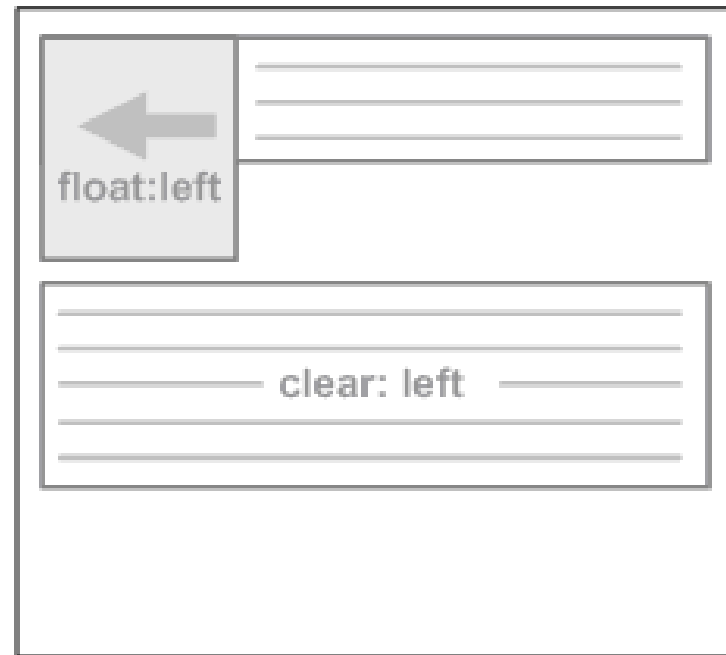
Estilos. Posicionamiento: clear

Elemento contenedor



Posicionamiento float de la imagen

Elemento contenedor

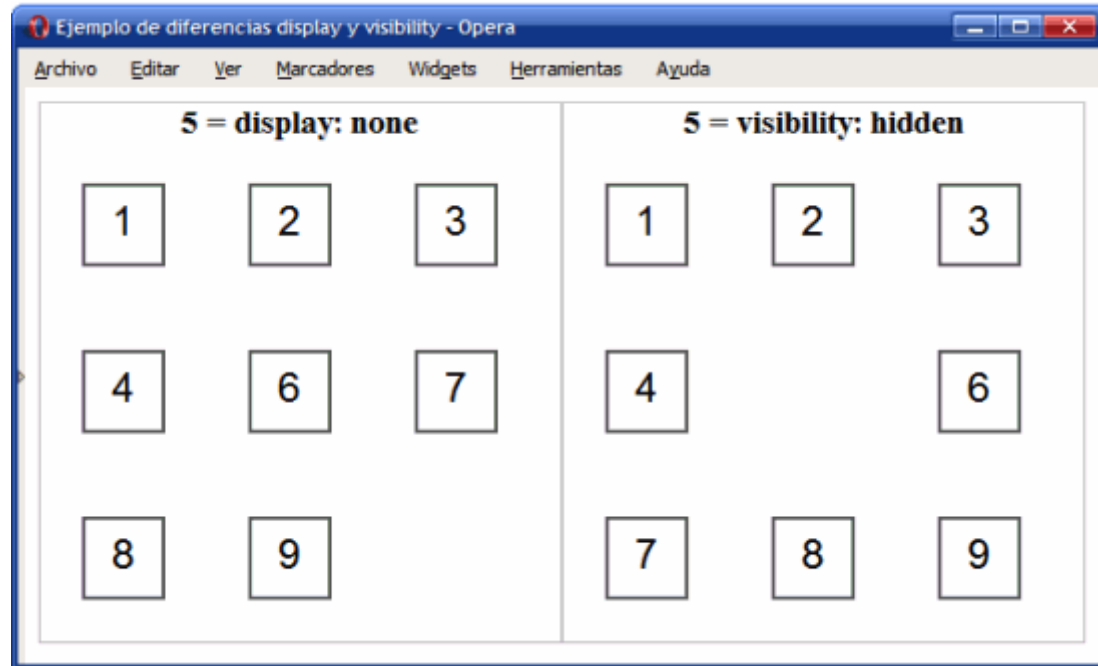


Párrafo "limpio" con clear:left

[Visualización]

- CSS define cuatro propiedades que definen la visualización de los elementos: display, visibility, overflow y z-index.
- Permiten ocultar y/o mostrar elementos.

Visualización: Display y visibility



- Diferencia entre display y visibility.

Visualización: 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 |

- Los más utilizados son inline, block y none.
- inline permite mostrar cualquier elemento como si fuera de línea.
- block permite mostrar cualquier elemento como si fuera de bloque.
- none hace como si el elemento no existiera

Visualización: visibility

| | |
|----------------------|---|
| 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 |

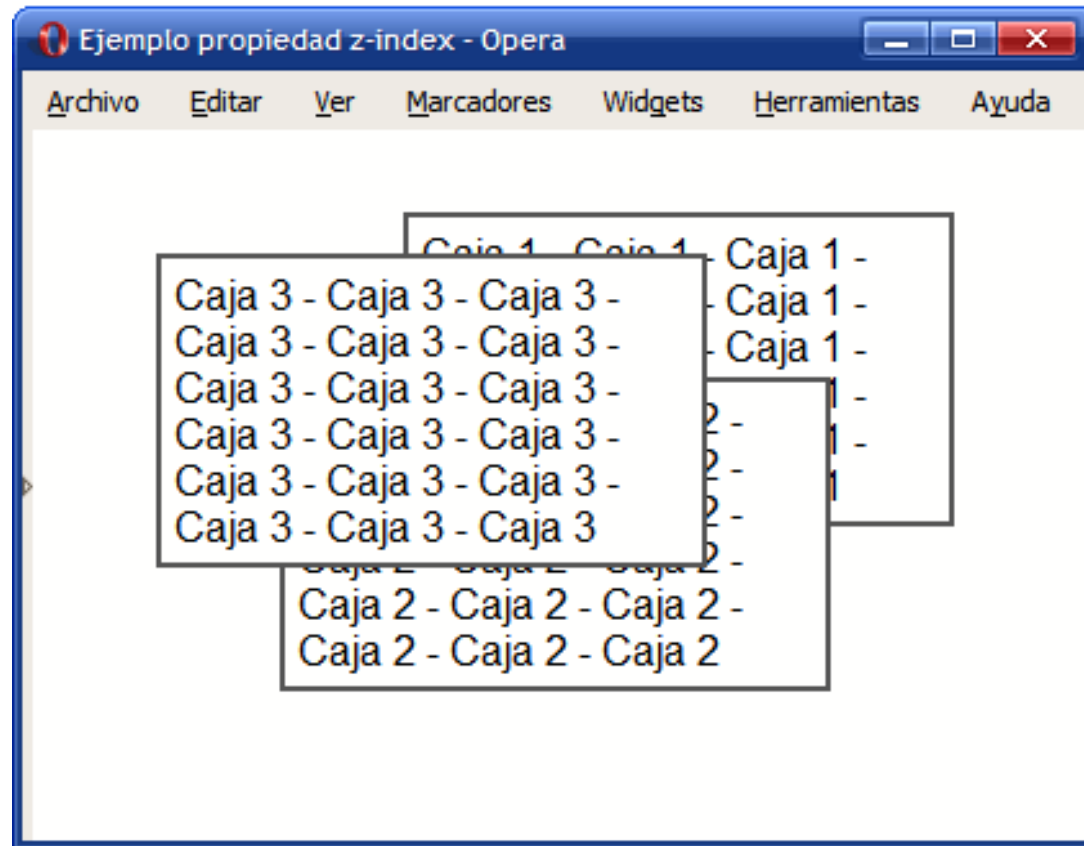
- Es posible hacer invisible a un elemento.
- Sin embargo, sigue “ocupando su lugar” para las maneras de posicionarse.

Visualización: z-index

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

- Valor tridimensional que indica que bloque está “detrás” de cual.
- Cuanto más alto es el número, más cerca del usuario se encuentra el elemento.

Visualización: z-index



Visualización: z-index

```
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>
```

[Vínculos]

- Introducción a CSS.
<http://www.librosweb.es/css/>
- Ejemplo de CSS:
<http://www.csszengarden.com/tr/espanol/>
- <http://www.csszengarden.com> (en inglés)