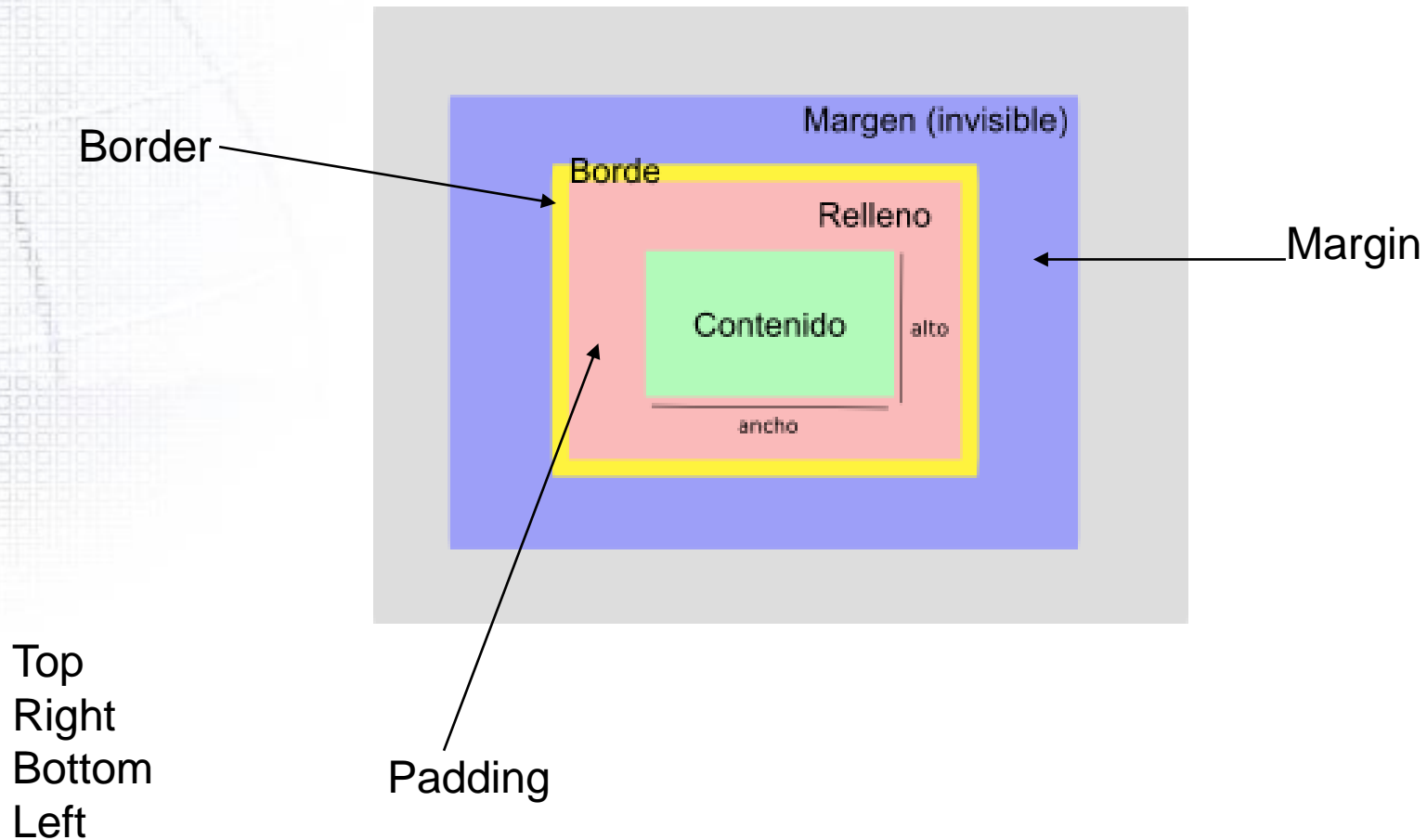




Modelo de caja

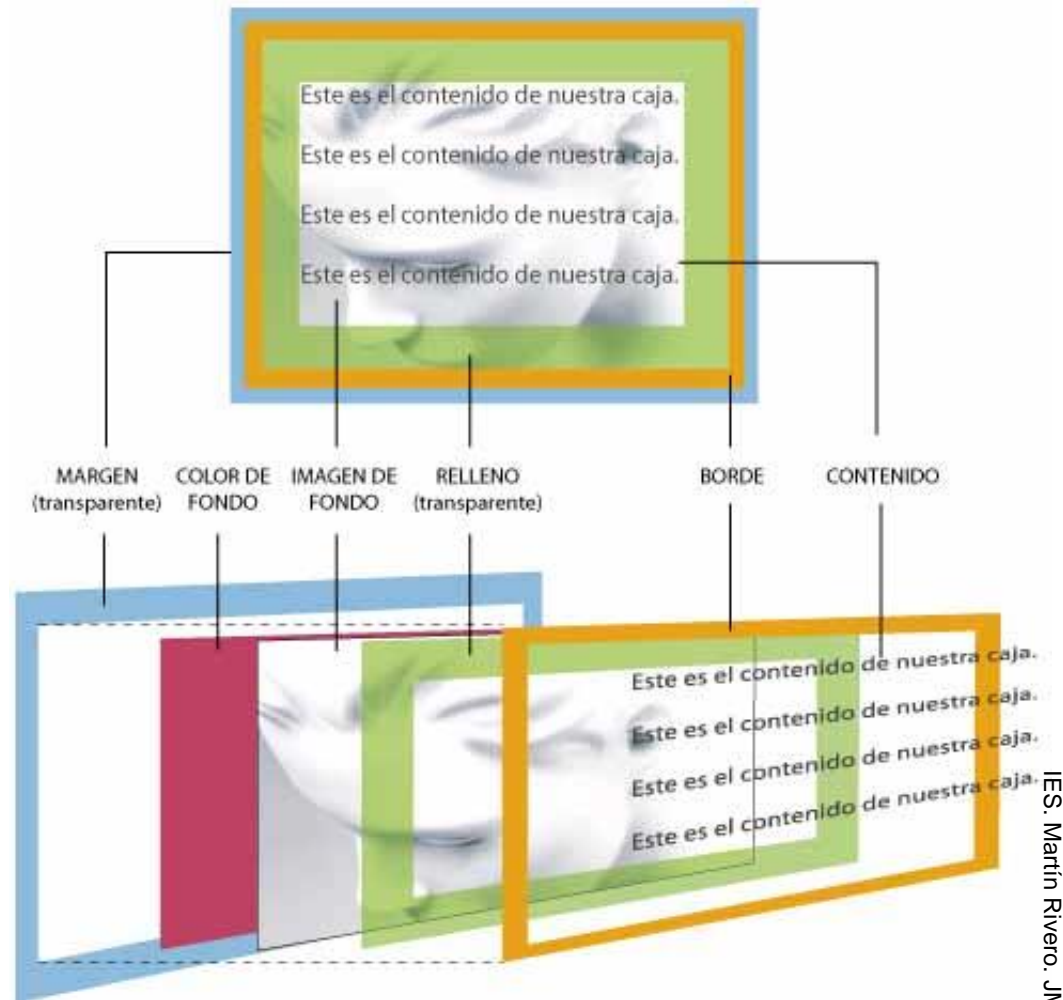
Uno de los pilares del CSS. Indica cómo se muestran los elementos.





Modelo de caja

- Relleno o Padding:
Se aplica alrededor del área de contenidos. Cuando añadimos un fondo a un elemento, se aplica al área formada por el contenido y el relleno.
- Márgenes:
Son Transparentes. Con ellos controlamos espaciado de los elementos. Aceptan valores negativos.



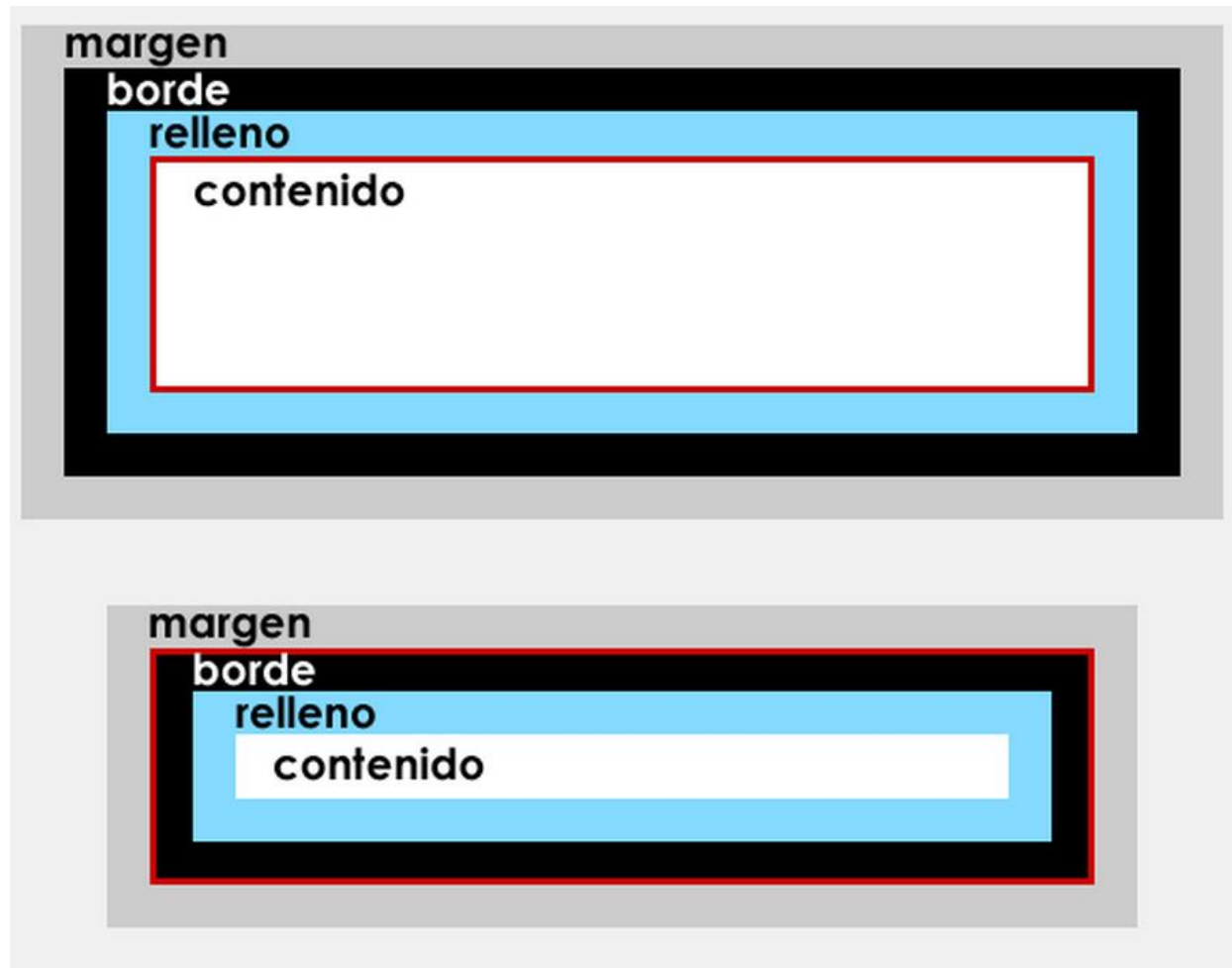
Relleno, bordes y márgenes, son opcionales y predeterminados en 0.



Modelo de caja

Podrían estas dos
cajas tener el mismo
valor de width?

Width:200px



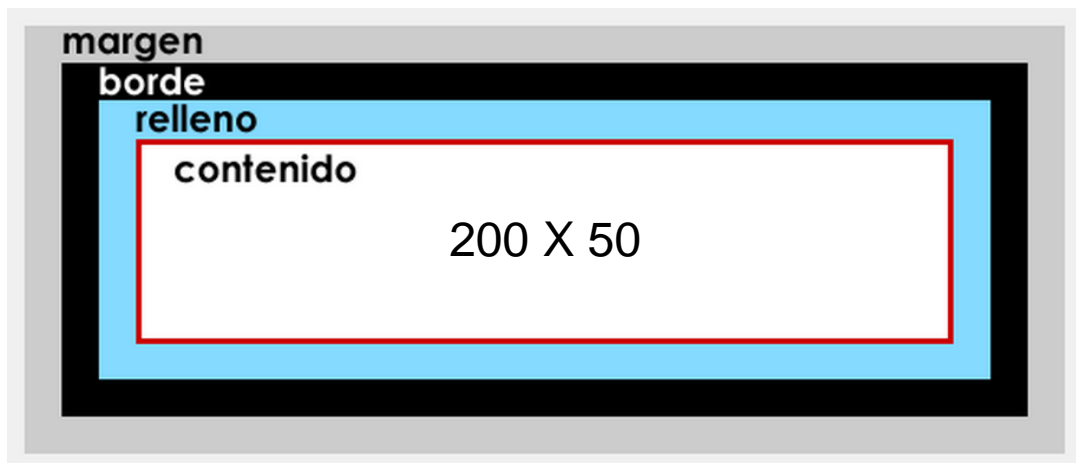


Modelo de caja - W3C

Según el W3C, el **ancho y alto** asignados a un elemento debe hacer referencia al área de **contenido**. A ese tamaño, se **añade** el espacio que se designe al relleno (*padding*) mas el borde (*border*). Al total, por último, se suma el tamaño del margen (*margin*), si es que éste se ha asignado.

Es decir, las cajas crecen de dentro a fuera. El modelo de caja del W3C se dice que es **aditivo**.

```
p{ width:200px;  
  height:50px;  
  padding:10px;  
  border:10px;  
  margin:10px; }
```



El área asignada al contenido será de 200 por 50 píxeles; a ésta se suma por cada lado 10 píxeles del relleno, 10 del borde y 10 del margen. El espacio asignado al elemento es, por tanto, de 260 por 110 píxeles.



Modelo de caja tradicional de IE

El modelo tradicional de Internet Explorer ha sido **sustractivo**: el espacio asignado a un elemento es el total resultante de sumar **contenido**, **relleno** y **borde**, es decir, que estas dos últimas propiedades reducen el espacio final asignado al contenido.

Crecen de fuera a dentro.

```
p{ width:200px;  
  height:50px;  
  padding:10px;  
  border:10px;  
  margin:10px; }
```



A los 200 píxeles iniciales se les añade los 10 del margen por cada lado, pero se les **resta** el **borde**, y el **relleno**. El área total del elemento resultante es de 220 por 70 píxeles, dentro de la cual el espacio del contenido se ve reducido a 160 por 10 píxeles.



Modelo de caja IE

Lógica de IE: Cuanto mas relleno, menos espacio quedará para el contenido

Decisión salomónica por parte del W3C:

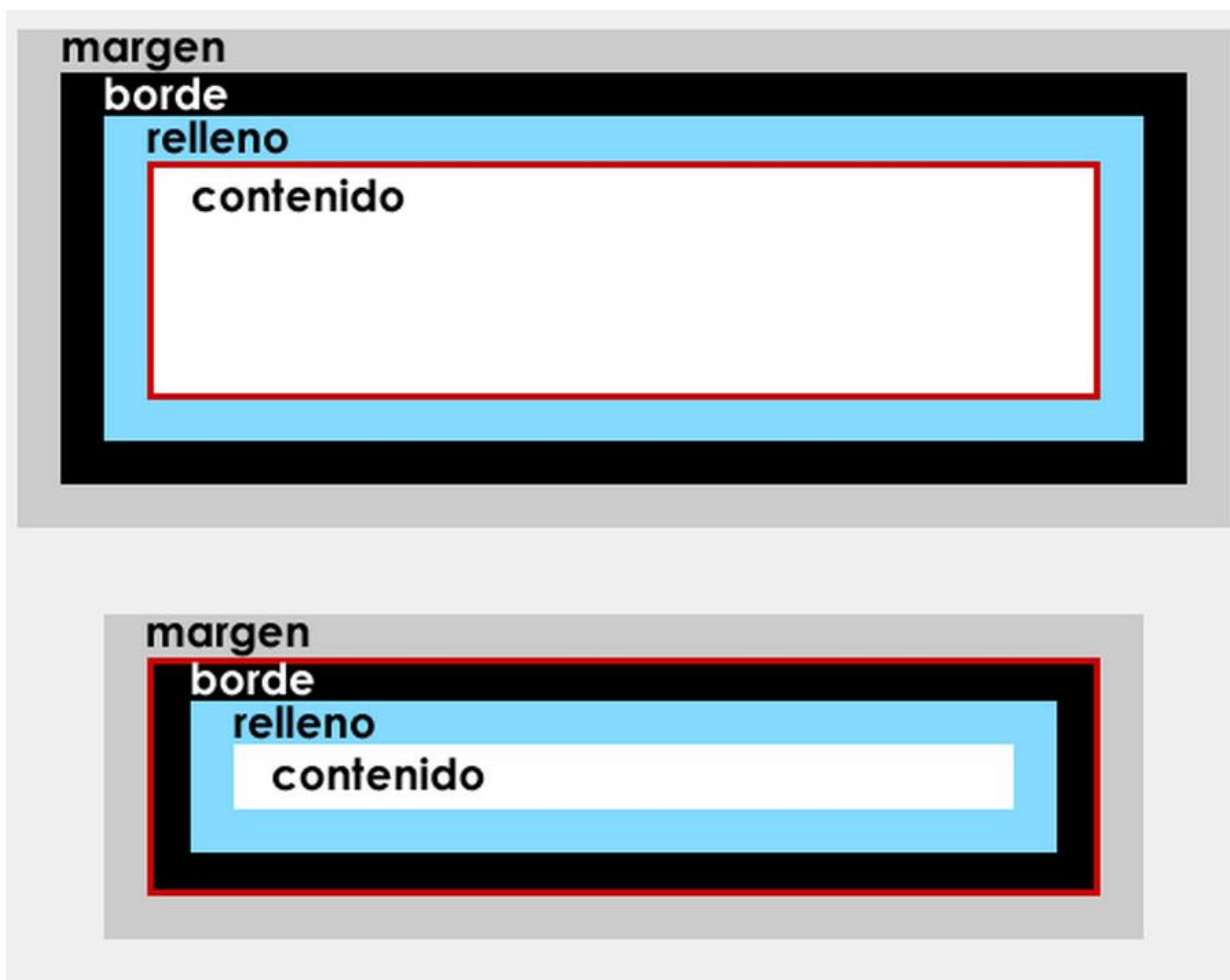
En CSS 3 se podrá elegir el modelo de caja a través de la propiedad **box-sizing**. Esta propiedad contará con los valores **content-box** para aplicar el modelo de la especificación 2.1, y **border-box** para emplear el modelo de Explorer.



Modelo de caja

Podrían estas dos
cajas tener el mismo
valor de width?

Width:200px

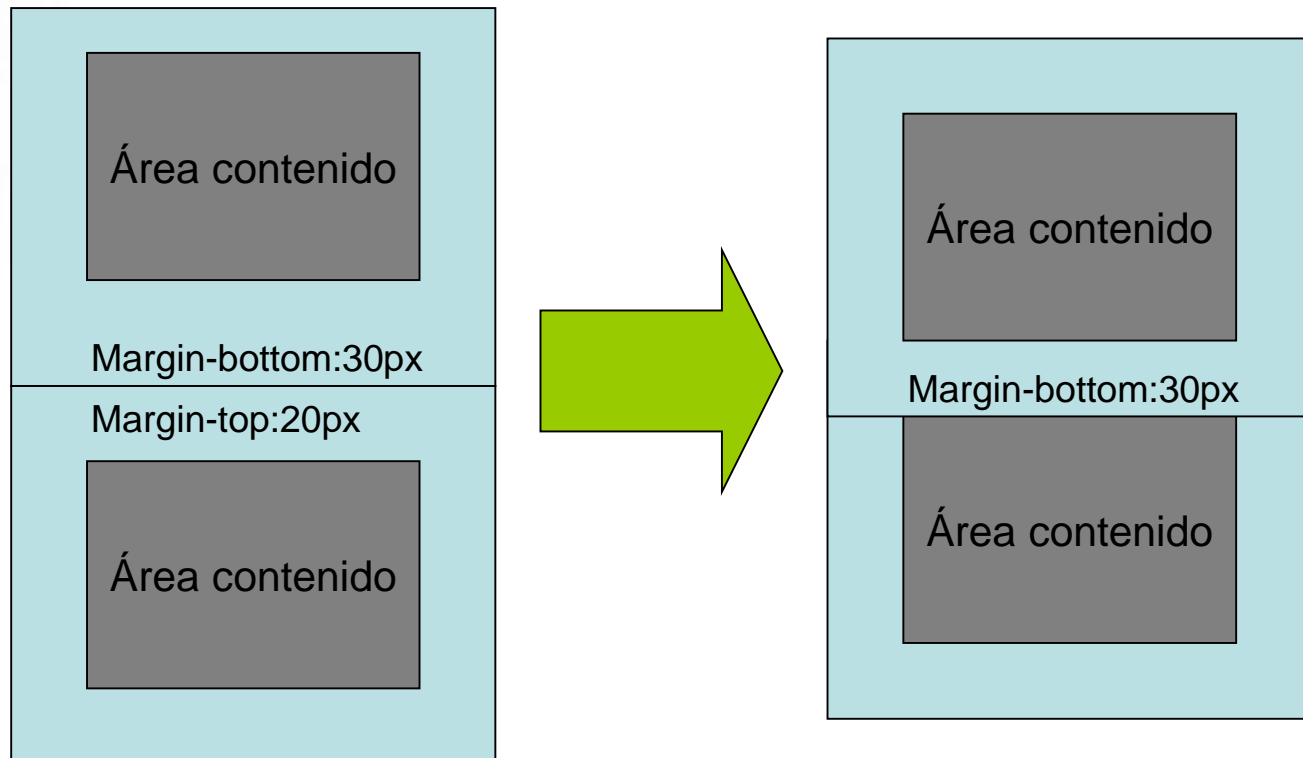




Combinar márgenes verticales

Genera mucha confusión. Cuando dos o más vértices verticales se encuentran, se combinarán para formar un solo margen. La altura de este será igual a la altura del MAYOR de los dos márgenes combinados.

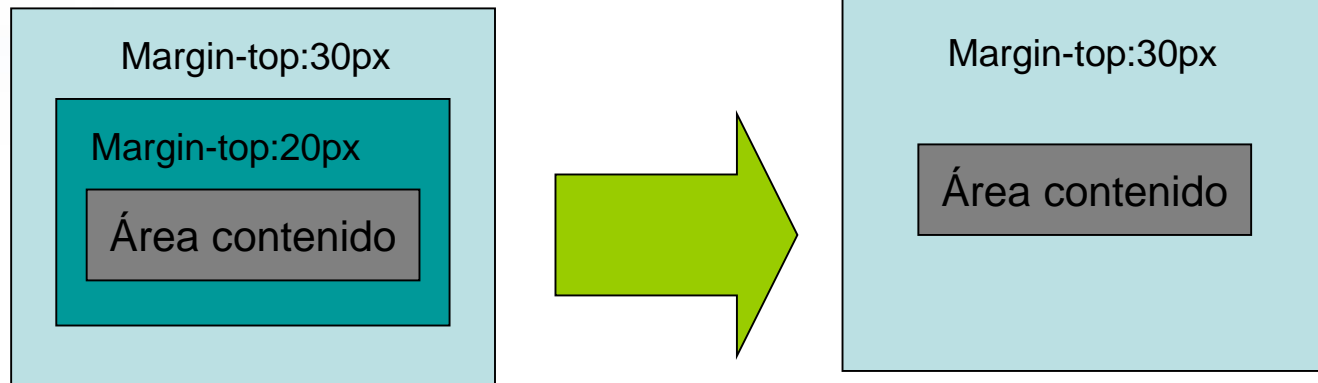
Cuando dos elementos están uno sobre otro, el margen inferior del primer elemento se combinará con el superior del segundo elemento.





Combinar márgenes verticales

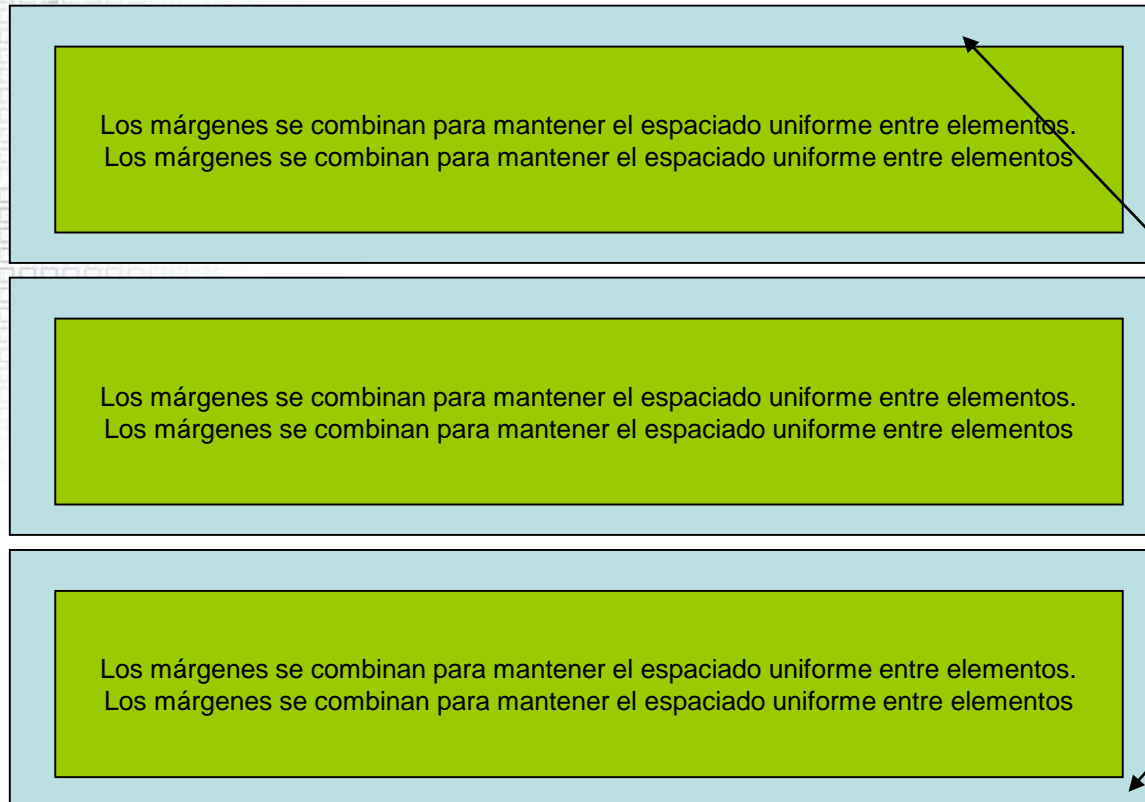
Cuando un elemento se encuentra dentro de otro elemento, suponiendo que **no** hay relleno o borde que separe los márgenes, sus márgenes superior y/o inferior también se combinarán juntos:





Combinar márgenes verticales

Los márgenes se combinan para mantener el espaciado uniforme entre elementos. Solo sucede con los márgenes verticales. Márgenes entre cajas en línea, flotantes o Posicionadas de forma absoluta, **nunca se fusionan**.



Margen superior e inferior
No tendrían la misma
altura que los interiores



Posicionamiento

Con ellos controlaremos la forma en que cada elemento se organiza en una página.

Modelo de Formato Visual

Existen elementos que se muestran visualmente como **bloques** (p, h1, div,...).

Existen elementos que se describen como elementos en línea (strong, span, a, ...), son las llamadas “**cajas de línea**”.

Con la propiedad “display”, podemos cambiar el tipo de caja que genera un elemento por defecto:

Ejemplo. Podemos hacer que un elemento en línea como un ancla, se comporte como un elemento a nivel de bloque al establecer su propiedad display a “block”. Si quiero que un elemento no genera caja, usaré display=“none”. La caja y todo su contenido, no se muestra y por tanto no ocupa espacio en el documento.





Posicionamiento

Existen 3 esquemas básicos de posición en CSS:

- Flujo normal (por defecto)
- Posicionamiento
- Flotación

Flujo normal: Según posición del elemento en XHTML. Según sea el tipo de caja (línea o bloque), el espaciado horizontal y vertical será diferente.





Posicionamiento Static.

Static

Muy parecido al flujo normal de los elementos. La capa se muestra donde le toque, según el código.

Los atributos **Left, right, bottom, top y z-index se ignoran.**



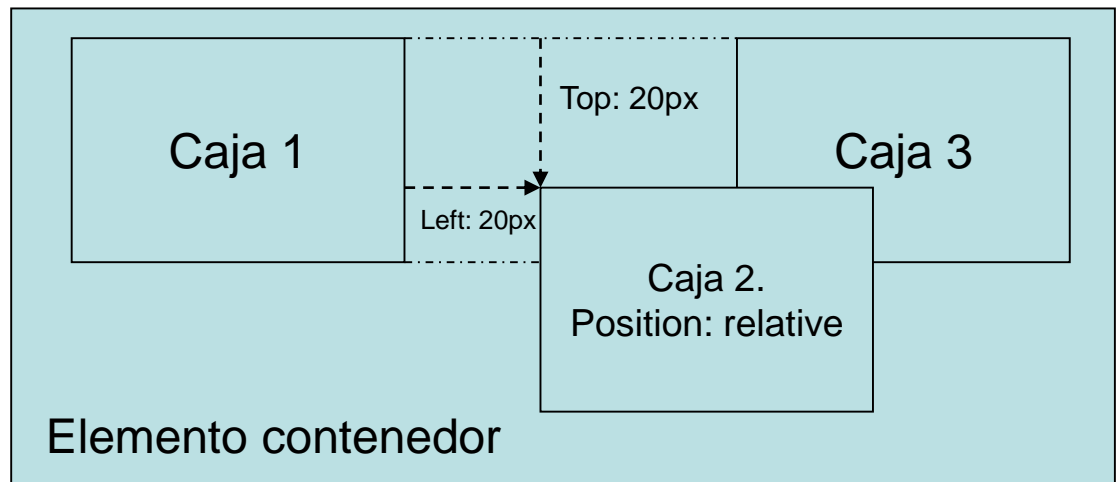
Posicionamiento relativo.

Los elementos posicionados relativamente, forman parte del flujo del documento. El origen de un elemento posicionado relativamente es el final del elemento previo en el flujo (en el flujo del documento, no tiene porque en el código fuente) independientemente de su relación jerárquica.

Para elementos con **posición relativa**, el origen de coordenadas es **la posición "natural"** que tendrían de no estar posicionados, en lugar de la esquina del elemento contenedor en la página.

El elemento continúa ocupando el espacio original, tanto si está separado como si no.

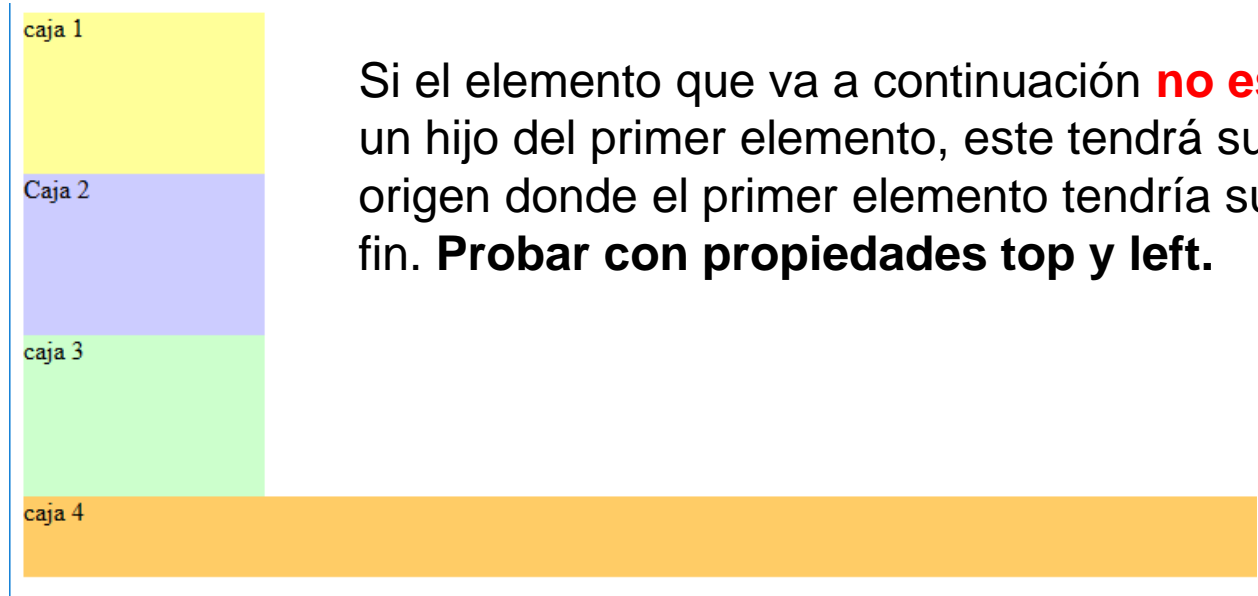
Por tanto, separar el elemento puede hacer que solape otras cajas.





Posicionamiento relativo. Ejemplo: 1

Cajas con
posicionamientos
relativos.



Si el elemento que va a continuación **no es** un hijo del primer elemento, este tendrá su origen donde el primer elemento tendría su fin. **Probar con propiedades top y left.**

```
.caja1, .caja2, .caja3, .caja4 {  
  height: 100px;  
  width: 150px;  
  background-color: #FF9;  
}  
.caja2 { background-color: #CCF }
```

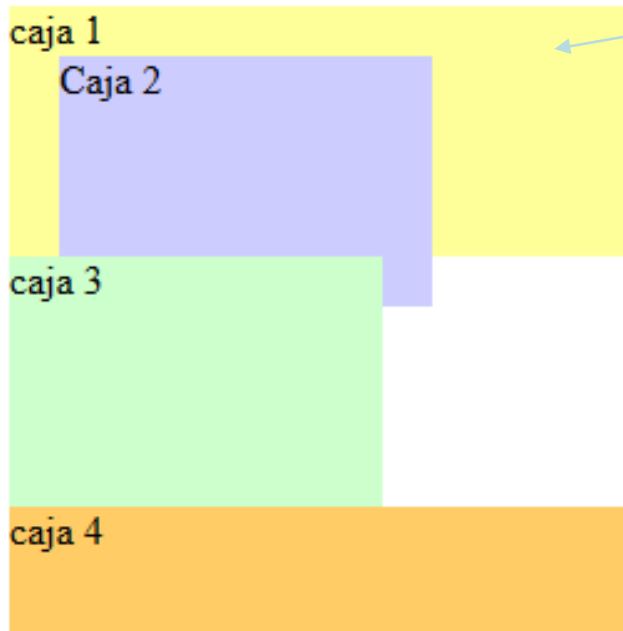
```
.caja3 { background-color: #CFC }  
.caja4 {  
  height: 50px;  
  width: 100%;  
  background-color: #FC6;  
}
```

position:relative; A CADA UNO



Posicionamiento relativo. Ejemplo: 2

Cajas 1 y 2 **Anidadas** con posicionamientos relativos.
El origen se establece donde acaba el contenido del **padre**.



```
<div class="caja1">caja 1  
<div class="caja2">Caja 2 </div></div>
```

```
.caja1 {  
  position:relative;  
  height: 100px;  
  width: 250px;  
}
```

```
.caja2 {  
  position:relative;  
  height: 100px; width: 150px;  
  left:20px;  
}
```

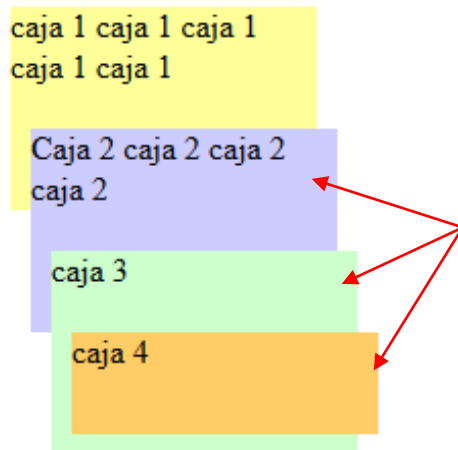


Posicionamiento relativo. Ejemplo: 3

En este ejemplo todas las cajas están anidadas. Separadas unas de otras por 20 pixels respecto al contenido de su padre. Si separamos caja padre se mueven sus hijos.

```
.caja1 {  
  height: 100px; width: 150px;  
  background-color: #FF9;  
}  
.caja2 {  
  position: relative;  
  background-color: #ccf;  
  height: 100px;  
  left: 10px;  
  top: 20px;  
  width: 150px; }
```

```
.caja3 {  
  position: relative;  
  height: 100px;  
  width: 150px;  
  background-color: #CFC;  
  left: 10px; top: 20px;  
}  
.caja4 {  
  position: relative;  
  height: 50px;  
  background-color: #FC6;  
  width: 100%;  
  left: 10px; top: 20px; }
```



Relativas

```
<div class="caja1">caja 1  caja 1 caja 1 caja 1 caja 1  
  <div class="caja2">Caja 2 caja 2  caja 2 caja 2  
    <div class="caja3">caja 3  
      <div class="caja4">caja 4  
    </div></div></div></div>
```



Posicionamiento Absoluto

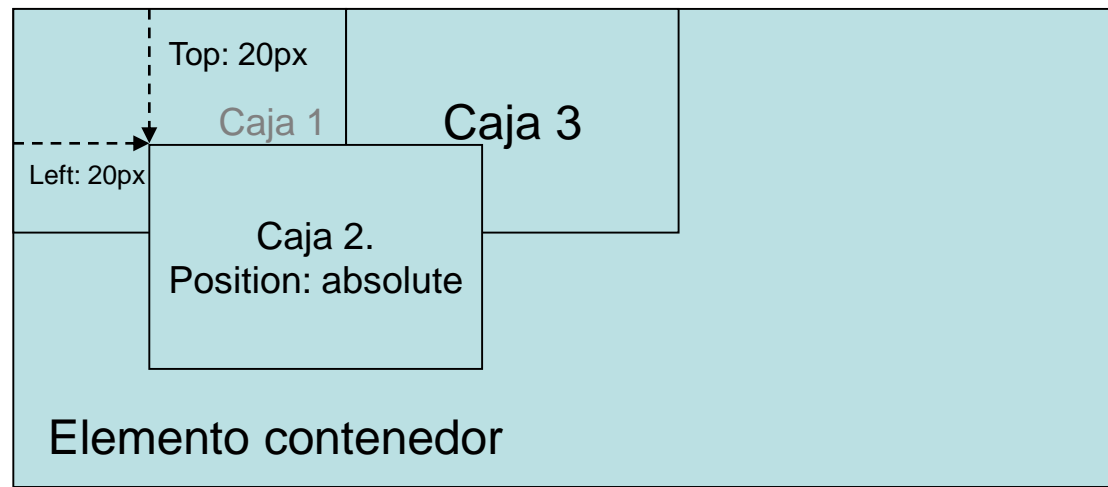
La posición absoluta, saca el elemento del flujo del documento, no ocupando ningún espacio. Otros elementos en el flujo normal del documento actuarán como si el elemento posicionado absolutamente nunca estuviera ahí.

Un elemento posicionado absolutamente se posiciona **en relación a su ancestro más cercano con posicionamiento establecido (relativo o absoluto)**. Si el elemento no tiene ancestros, se posicionará en relación al bloque de contenido inicial (body).

Flexibilidad. Podemos posicionar literalmente un elemento en cualquier parte de la página.

Pueden solapar otros elementos.

Podemos controlar el orden con propiedad **z-index=x**, donde x será un número entero positivo. (A mayor valor de x, más arriba aparecerá la capa).





Posicionamiento Absoluto. Ejemplo: 4

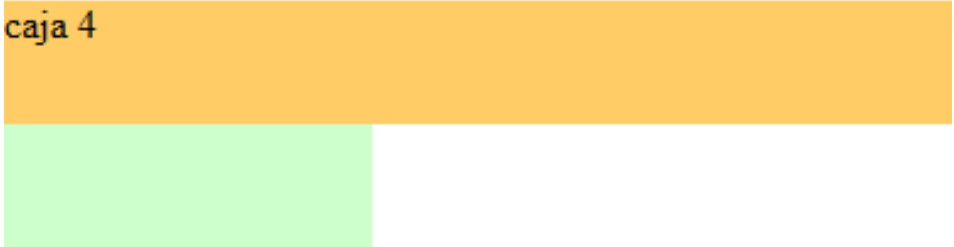
Todas las capas con posicionamiento **absoluto sin anidar**. Se apilan y solo es visible al última. Su ancestro mas cercano será el body. **No hemos fijado coordenadas de origen.(top, left...)**, luego se colocarán a continuación del elemento que las preceda en código.

```
.caja1 {  
    position: absolute;  
    height: 100px; width: 150px;  
    background-color: #FF9;  
}  
  
.caja2 {  
    position: absolute;  
    background-color: #ccf;  
    height: 100px; width: 150px;  
}
```

```
.caja3 {  
    position: absolute;  
    height: 100px; width: 150px;  
    background-color: #CFC;  
}  
  
.caja4 {  
    position: absolute;  
    height: 50px; width: 100%;  
    background-color: #FC6;  
}
```

Probar añadiendo a alguna capa la propiedad top. La capa en cuestión se posicionará con respecto al body.

caja 4





Posicionamiento Absoluto. Ejemplo: 5

Todas las capas con posicionamiento absoluto y **anidadas**. Se apilan pero su posición de inicio se fija con respecto al **contenido** de su padre. Su ancestro mas cercano ya no es el body, será la capa respectiva en la que se encuentre. Si usamos top y left en caja 2 se moverán todas las demás.

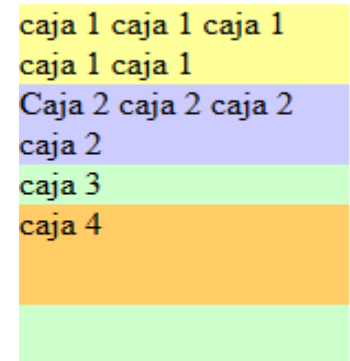
```
.caja1 {  
    position: absolute;  
    height: 100px; width: 150px;  
    background-color: #FF9;  
}
```

```
.caja2 {  
    position: absolute;  
    background-color: #ccf;  
    height: 100px; width: 150px;  
}
```

```
.caja3 {  
    position: absolute;  
    height: 100px; width: 150px;  
    background-color: #CFC;  
}
```

```
.caja4 {  
    position: absolute;  
    height: 50px; width: 100%;  
    background-color: #FC6;  
}
```

```
<div class="caja1">caja 1  caja 1 caja 1 caja 1 caja 1  
<div class="caja2">Caja 2 caja 2  caja 2 caja 2  
<div class="caja3">caja 3  
<div class="caja4">caja 4 </div></div></div></div>
```



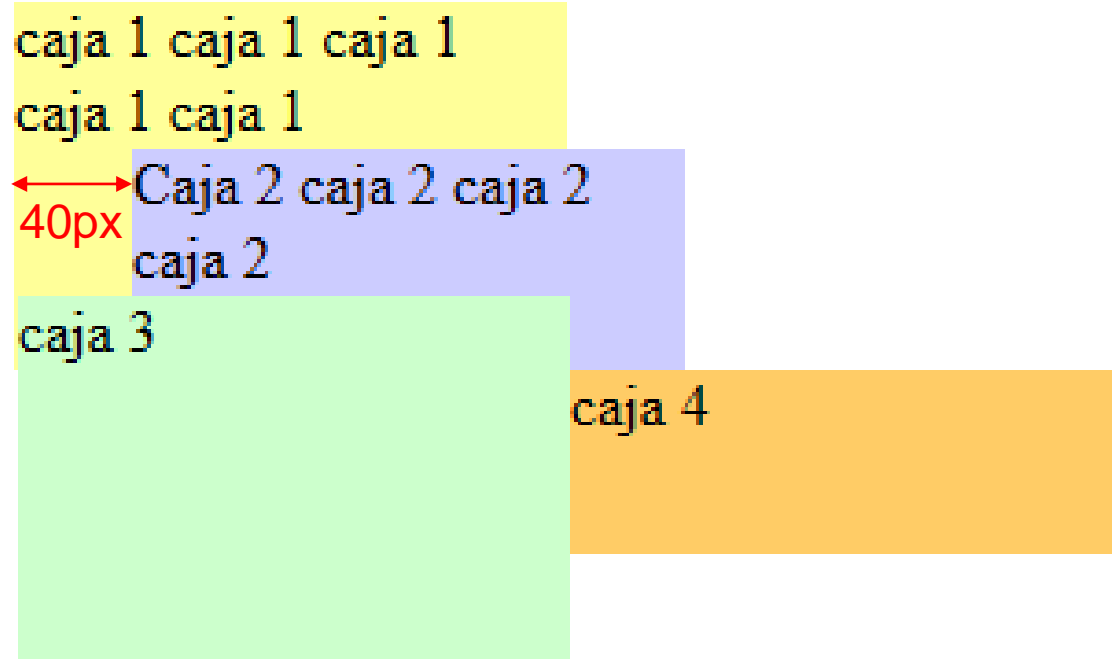
Probar asignando a capa 4 la propiedad top y anulando los posicionamientos de sus ancestros.

Posicionamiento Absoluto. Ejemplo: 6

Conseguir con el ejemplo anterior una disposición de cajas así:

Quitar posicionamiento a caja 1.

Hacer que caja 3 empiece en margen izq. Como caja 1.

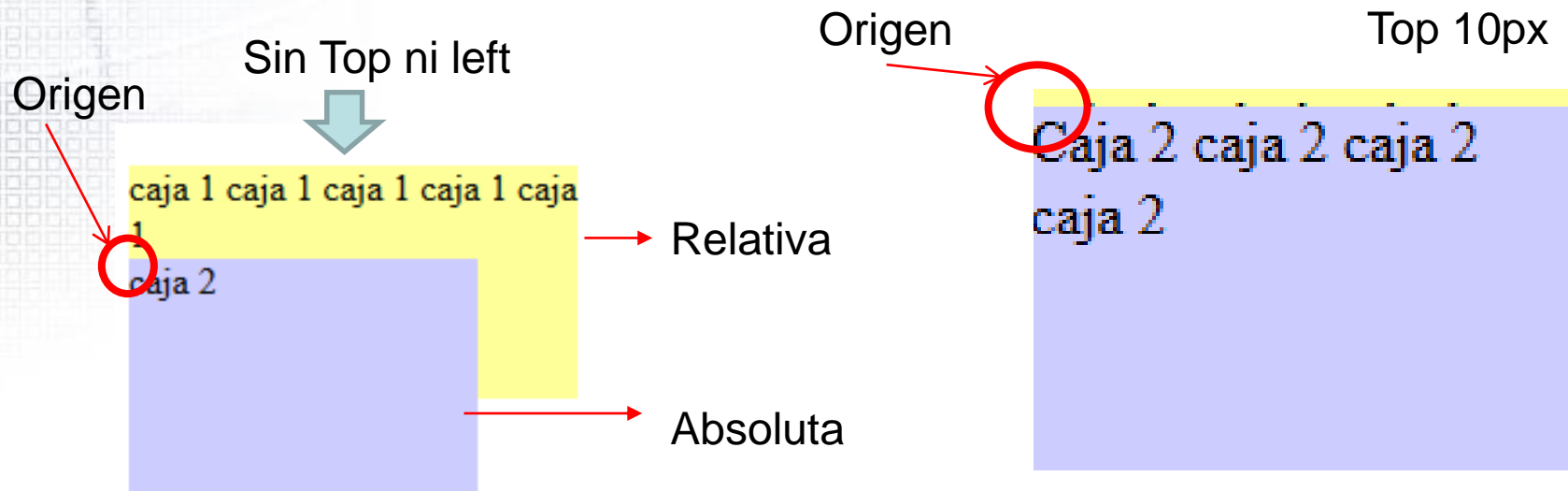




Posicionamientos Absolutos - Relativos. Ejemplo: 7

Dos elementos **anidados** posicionados de forma Relativa y absoluta. Movimientos de caja absoluta:

1. Si en la capa absoluta, se **omite** la propiedad **top**, o se le asigna valor *auto*, la parte superior de nuestro elemento absoluto comenzará a continuación de la última línea del elemento padre. (Con left pasará algo parecido)
2. Si **especifico top o left**, el origen se referenciará desde el principio de la caja de su ancestro.



Si caja 1 fuese Static, la capa 2 absoluta la ignoraría completamente y se fijaría con respecto al body (su ancestro mas cercano).



Posicionamiento Absolutos - Relativos. Ejemplo: 8

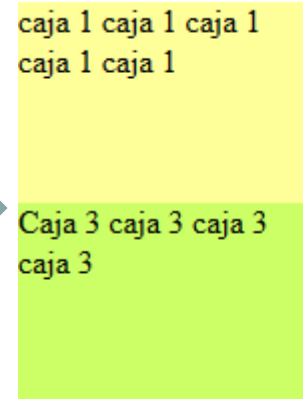
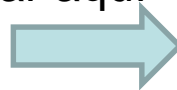
Si tenemos 3 elementos **no anidados**: Relativo – Absoluto – Relativo. El absoluto aunque esté declarado en código detrás del relativo, será como si flotase. No influye en los relativos.

```
.caja1 {  
  position: relative;  
  height: 100px;  
  width: 150px;  
  background-color: #FF9;  
}
```

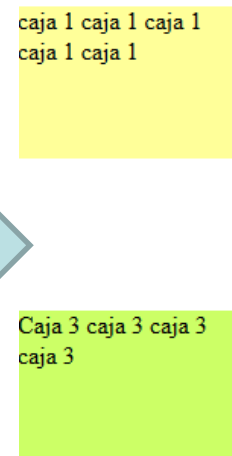
```
.caja2 {  
  position: absolute;  
  background-color: #ccf;  
  height: 100px;  
  width: 150px;  
  top: 160px;  
  left: 200px;  
}
```

```
.caja3 {  
  position: relative;  
  height: 100px;  
  width: 150px;  
  background-color: #CF6;  
}
```

Debería
estar aquí



Si caja2 fuese
relativa, se reserva
el hueco





Posicionamiento Fixed.

El posicionamiento fijo (fixed) es una subcategoría del posicionamiento absoluto. Las cajas fixed están fuera del flujo del documento.

La única **diferencia** con el posicionamiento absoluto es que su posición respecto del bloque de contención es establecido por el acceso visual (pantalla, pagina impresa, etc). Para los medios continuos (pantallas), los bloques en posición fixed **no se mueven cuando el documento es desplazado**. En este aspecto, son similares a las imágenes de fondo fijas, y son útiles para establecer cabeceras o pies de página. Para los medios paginados, los bloques con posiciones fijas (fixed) son repetidos en cada pagina. Esto resulta útil para poner, por ejemplo, una firma al pie de cada página.

IE 6 e inferior, no soportan el posicionamiento fijo sin el doctype apropiado.



Juan Manuel Márquez Rubio
IES Martín Rivero 2017