

Posicionamientos

CSS

Estructura del BODY

<div id="example">

<div id="div-before">

<p>id = div-before</p>

</div>

<div id="div-1">

<div id="div-1-padding">

<p>id = div-1</p>

<div id="div-1a">

<p>id = div-1a</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Integer pretium dui sit amet felis. Integer sit amet diam. .</p>

</div>

<div id="div-1b">

<p>id = div-1b</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. dui sit
amet felis. Integer sit amet diam. Phasellus nisi tincidunt tortorcum
san augue ipsum id lorem.</p>

</div>

<div id="div-1c">

<p>id = div-1c</p>

</div>

</div>

</div>

<div id="div-after">

<p>id = div-after</p>

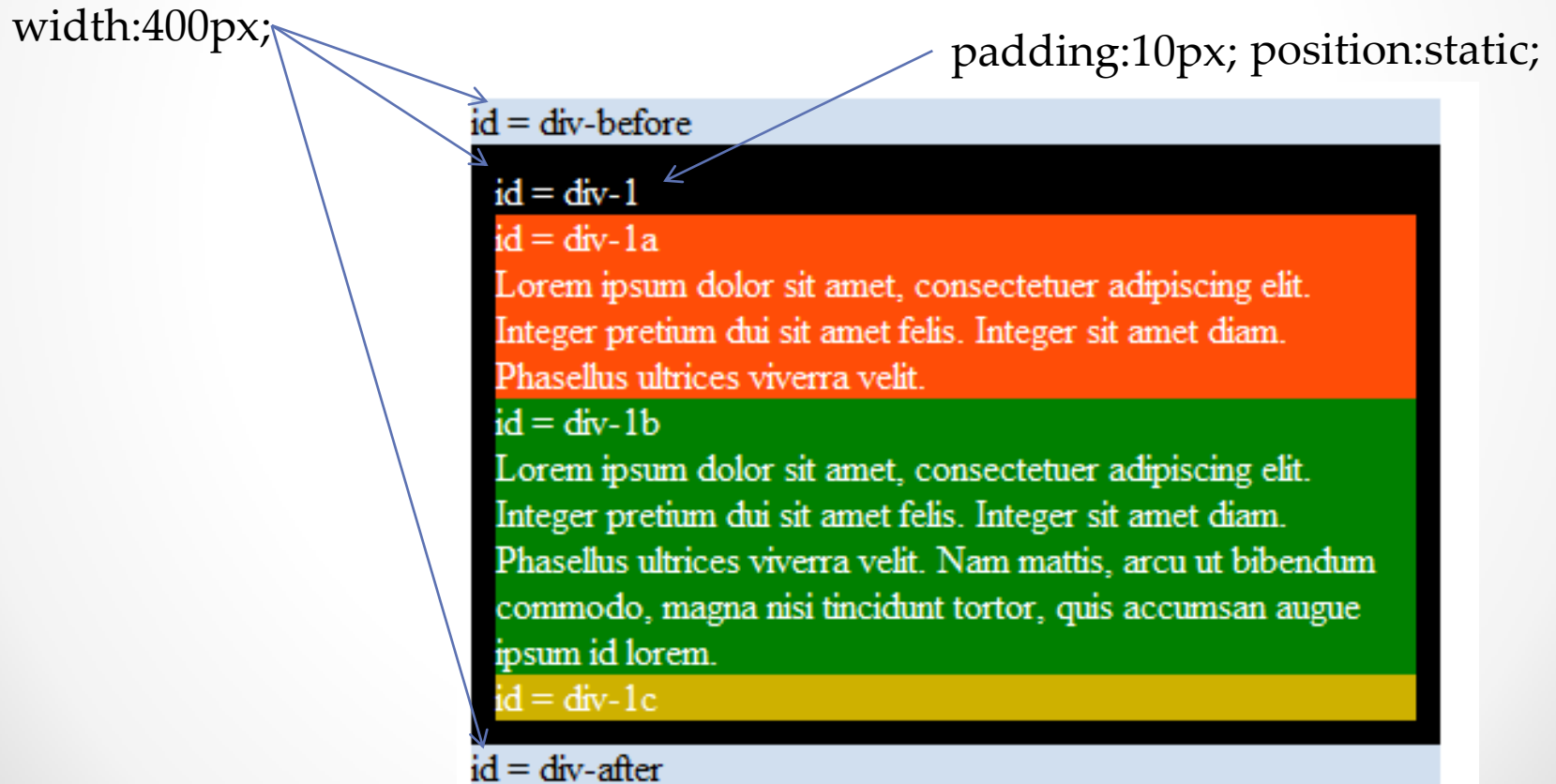
</div>

</div>

Posicionamiento **STATIC**

El posicionamiento **normal** de todos los elementos es **static**, que significa que cada elemento se posiciona donde le corresponde según el flujo normal de la página. Si no se indica nada, este es el valor por defecto que toman todos los elementos.

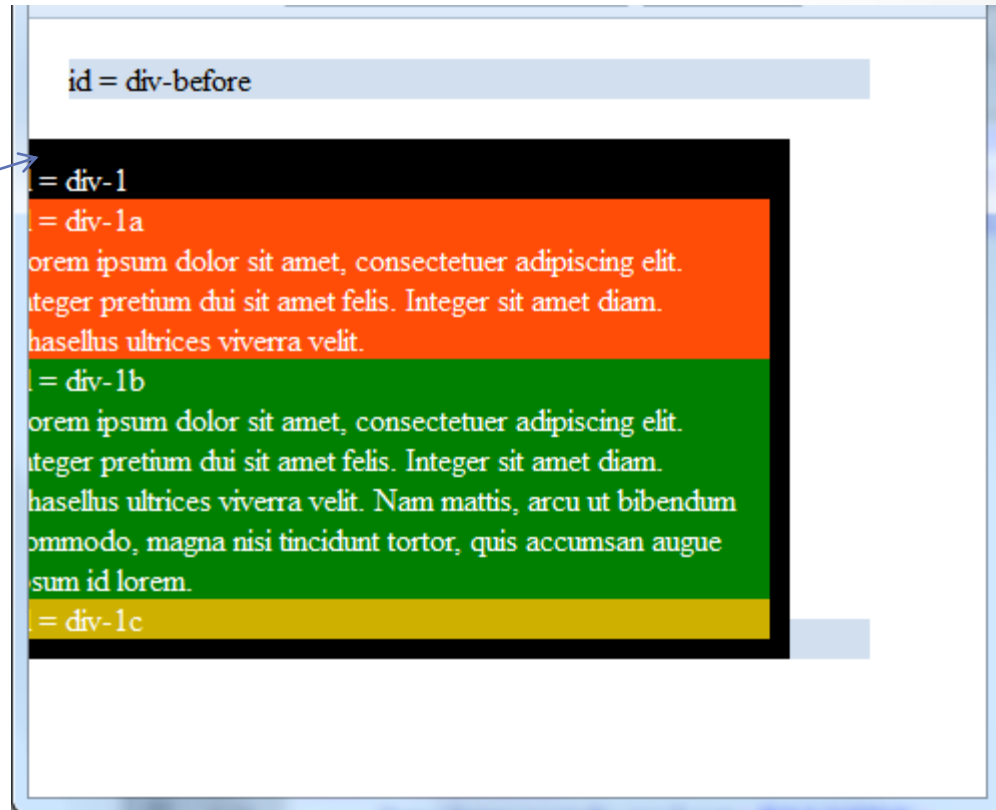
Los atributos **Left, right, bottom, top** y **z-index** se **ignoran**.



POSITION RELATIVE

Si se especifica la posición relativa para un elemento, puedes posicionar este elemento usando las propiedades **top**, **bottom**, **left** y **right**, y el elemento se posicionará a la distancia indicada respecto de donde debería estar según el flujo normal de la página.

```
#div-1 {  
  position:relative;  
  top:20px;  
  left:-40px; }
```

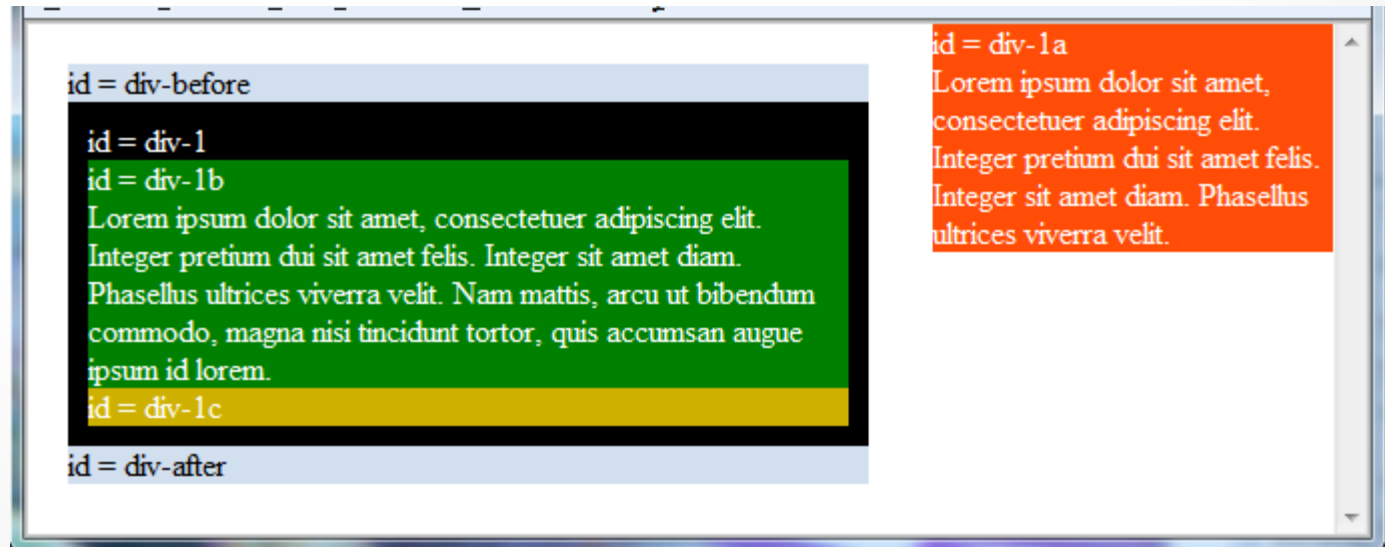


POSITION ABSOLUTE

Si se especifica position **absolute**, el elemento se muestra exactamente donde se indica en las propiedades **top**, **bottom**, **left** o **right**.

```
#div-1a {  
  position:absolute;  
  top:0;  
  right:0; ←  
  width:200px; }
```

Siempre estará a la derecha

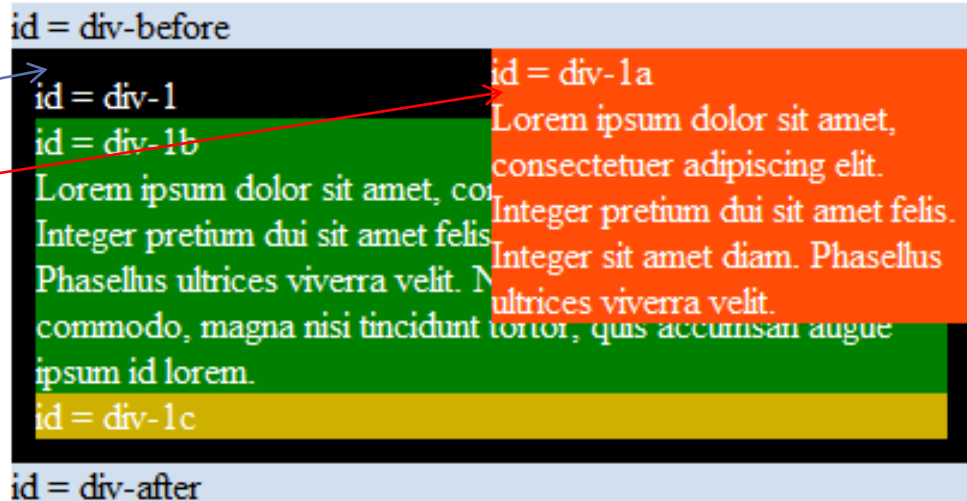


POSITION RELATIVE + POSITION ABSOLUTE

Si especificas la posición de div-1 de forma relativa, todos los elementos contenidos dentro de div-1 se posicionaran relativamente a div-1, por lo tanto si le establecemos la posición absoluta a div-1a, este se posicionará en la esquina superior derecha del contenedor div-1.

```
#div-1 {  
  position: relative; }
```

```
#div-1a {  
  position: absolute;  
  top:0;  
  right:0;  
  width:200px; }
```

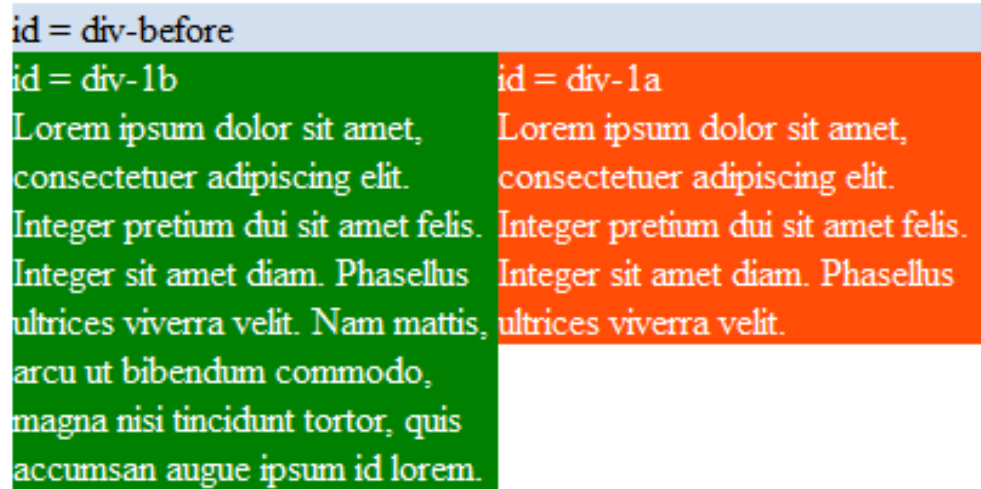


2 COLUMNAS POSICIONADAS DE FORMA ABSOLUTA

Utilizando posicionamiento relativo en el contenedor y absoluto para las columnas se pueden posicionar las columnas, una en cada lado.

Una ventaja de usar este posicionamiento es que se puede colocar el HTML de las columnas en cualquier orden, ya que el posicionamiento se realiza de forma absoluta.

```
#div-1 { position:relative; }
#div-1a {
    position:absolute;
    top:0;
    right:0;
    width:200px; }
#div-1b {
    position:absolute;
    top:0;
    left:0;
    width:200px; }
```

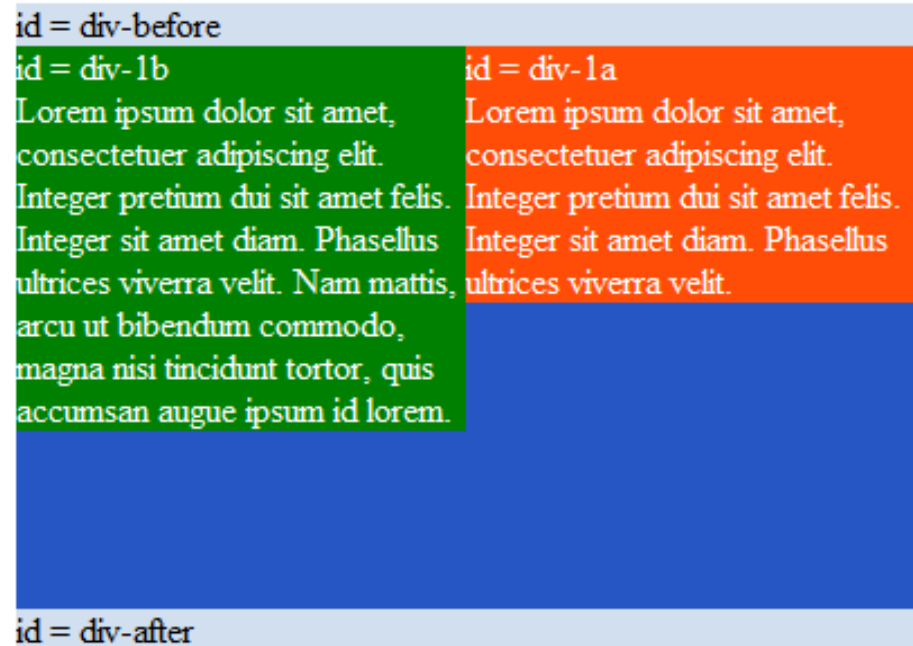


Los demás elementos han sido tapados por las dos columnas que están posicionadas de forma absoluta. Vamos a ver cómo solucionar esto.

COLUMNAS POSICIONADAS DE FORMA ABSOLUTA CON ALTO DEFINIDO

Para poder visualizar el contenido tapado por las dos columnas, habría que darle un alto definido al elemento contenedor, de esta forma ya se visualizarían.

```
#div-1 {  
    position:relative;  
    height:250px; }  
#div-1a {  
    position:absolute;  
    top:0;  
    right:0;  
    width:200px; }  
#div-1b {  
    position:absolute;  
    top:0;  
    left:0;  
    width:200px; }
```



Pero esta **no es una solución válida** para la mayoría de los diseños, porque **lo normal** es que **no se sepa** cual va a ser la cantidad del texto introducido en las columnas. ¿Cómo se puede solucionar esto?

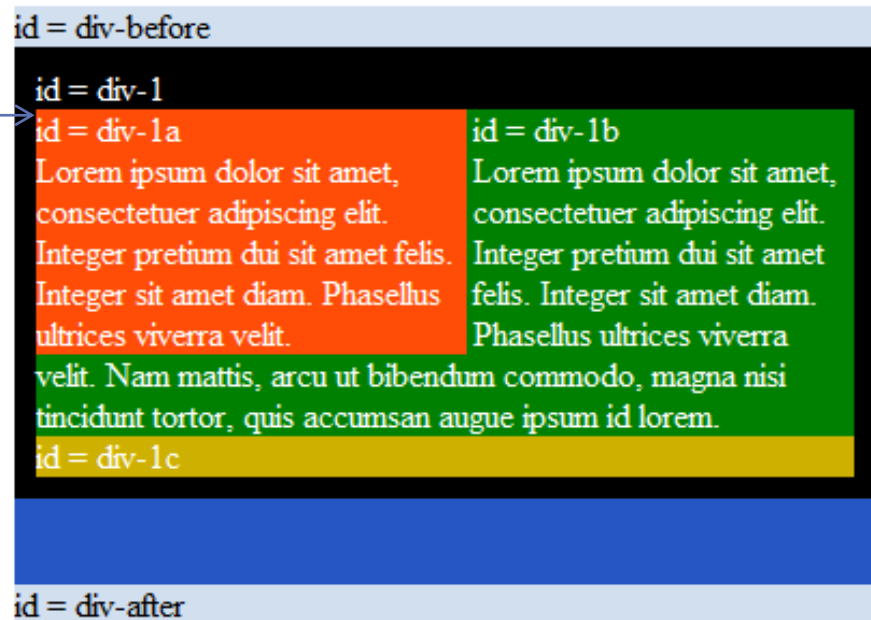
7-posi relative-absolute absolute con alto.html

POSICIONAMIENTO FLOTANTE

La **primera** solución al ejemplo anterior es utilizar el posicionamiento **flotante**. Con la propiedad float se puede posicionar un elemento lo más a la derecha posible o lo más a la izquierda posible, dejando que el resto de elementos fluya alrededor suyo.

El div **div-1a** está flotado a la izquierda, el texto del resto de divs **fluyen alrededor** de este. El resto de divs (div-1b y div-1c) siguen ocupando la misma posición que antes, es decir todo el contenedor, porque son elementos de bloque. Esto significa que **el texto no se solapará** con div-1, pero si le asignas una **imagen de fondo** con CSS a div-1b, esta imagen **sí** que se solapará con div-1a.

```
#div-1a {
    float:left;
    width:200px;
}
```



POSICIONAMIENTO FLOTANTE

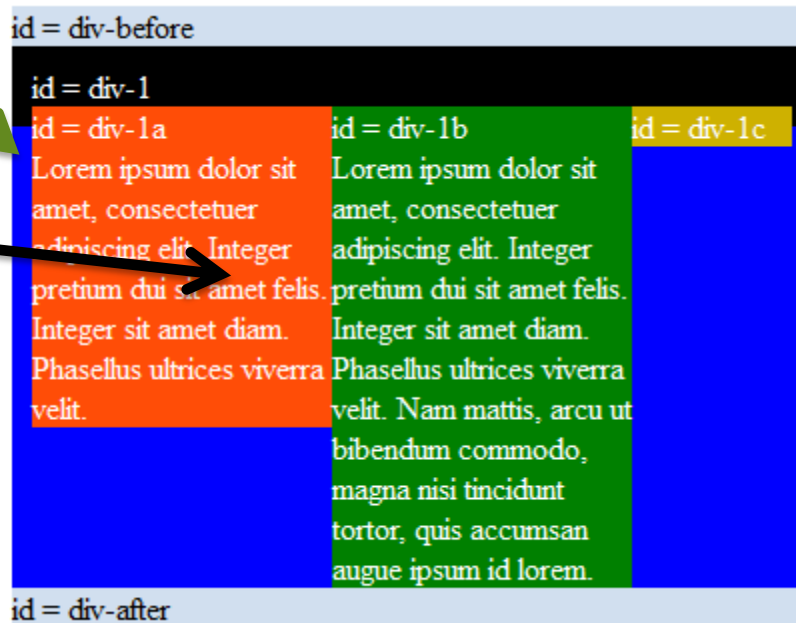
La segunda solución sería **flotar todas** las capas y su capa contenedora.

overflow: hidden; —————> Otra posibilidad

```
#div-1 { background:blue;
float:left; }
```

```
#div-1a, #div-1b {
float:left;
width:200px;
}
```

```
#div-1c {
float:left;
width:80px;
}
```



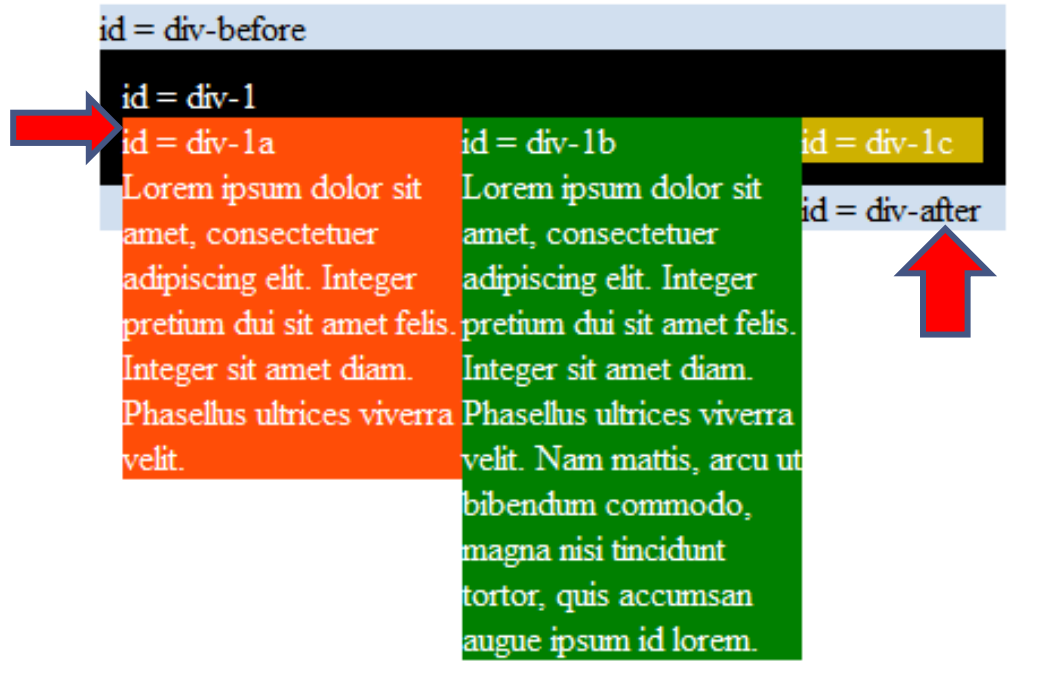
8-B posi flotante (Solución 2)

COLUMNAS FLOTANTES

Para que **no se solapen** unos **elementos** con otros hay que flotar todas las columnas. Para que se muestren una al lado de otra, entre todas las columnas han de sumar un ancho no superior al ancho del contenedor. Fíjate que, tanto div-after, como div-1 (que hace de contenedor) se solapan con las columnas creadas, vamos a ver como solucionar esto.

```
#div-1a {  
    float:left;  
    width:150px; }
```

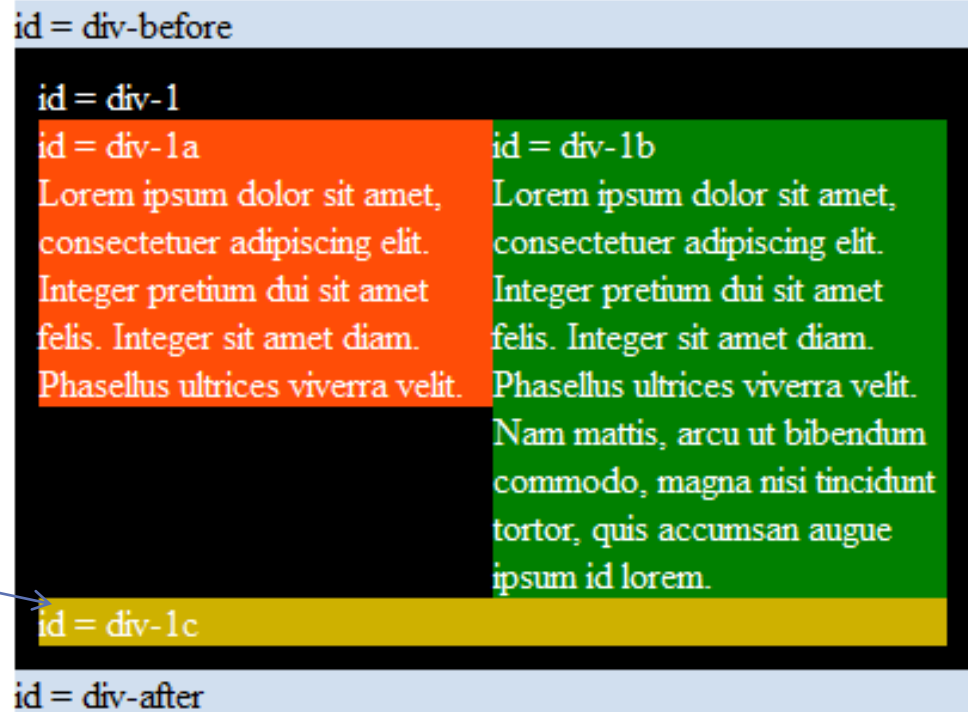
```
#div-1b {  
    float:left;  
    width:150px; }
```



COLUMNAS FLOTANTES CON CLEAR

Para solucionar el problema anterior, en el último div (div-1c), el que hace de pie, se le puede añadir la propiedad **clear**, para que deje de tener efecto el flotado de las columnas de arriba y el resto del contenido se posicione debajo.

```
#div-1a {  
    float:left;  
    width:190px; }  
#div-1b {  
    float:left;  
    width:190px; }  
#div-1c {  
    clear:both; }
```



10-posi columnas flotantes con clear