

# Flotar e posicionar

## Sumario

- 1 Introducción
- 2 Fluxo normal
- 3 Flotar
- 4 **Nota** Cando queremos flotar un elemento éste terá que ter definido implícita o expresamente o seu largo.
  - ◆ 4.1 Caixas non flotantes
  - ◆ 4.2 Flotar unha caixa á dereita
  - ◆ 4.3 Flotar unha caixa á esquerda
  - ◆ 4.4 Flotar todas as caixas á esquerda
- 5 **Nota** Tal e como se ve na imaxe anterior, o div "contido" ten agora unha altura de 0px, xa que todo o que ten no seu interior está fóra do fluxo do documento. É dicir, non ten nada dentro e non precisa ter unha altura determinada.

Para que se volva a ver o "contedor":

  - Pódesele poñer unha altura configurando no selector #contido a propiedade *height* cunha altura maior que o alto da caixa máis grande.

Así e todo os divs de cores seguen estando no interior do div "contido" e iso vese en que non saen do ancho marcado por este e o aliñamento é con respecto a el.
  - Tamén se pode flotar á esquerda a caixa con id #contido, así volvería a aparecer, iso si, flotada á esquerda.
  - Engadir un div totalmente baleiro dentro do div "contido" e despois das tres caixas flotadas e que configuraremos coa propiedade *clear* có valor, por exemplo, *both*. Esta é, sen dúbida, a mellor solución.
- 6 Marxes negativos e solapamento
- 7 Limpar (*clear*)
- 8 Posicionamento
  - ◆ 8.1 Introducción ao posicionamento
  - ◆ 8.2 Propiedade *display*
  - ◆ 8.3 Propiedade *position*
- 9 **Determinar orixe coordenadas** Determinar a orixe de coordenadas a partir do cal se despraza unha caixa situada de forma absoluta é un proceso complexo que se compón dos seguintes pasos:
  - ◆ Búscanse todos os elementos colectores da caixa até chegar ao elemento da páxina.
  - ◆ Percórrense todos os elementos colectores empezando polo máis próximo á caixa e chegando até o BODY
  - ◆ De todos eles, o navegador queda co primeiro elemento colector que estea situado de calquera forma diferente a *position: static*
  - ◆ A esquina superior esquerda dese elemento colector situado é a orixe de coordenadas.
  - ◆ 9.1 Propiedade *visibility*
  - ◆ 9.2 Propiedade *clip*
- 10 **Nota** Non afecta o espazo ocupado polo elemento no fluxo HTML.
- 11 **Nota** Para que o clip funcione a posición debe axustarse a *absolute*.
  - ◆ 11.1 Propiedade *overflow*
  - ◆ 11.2 Propiedade *z-index*
- 12 **Nota** Os elementos con maior Z-INDEX son colocados encima dos que teñan menor Z-INDEX , quedando estes últimos tapados polos primeiros. A propiedade Z-INDEX só se aplica a elementos que teñan a propiedade *position* en *absolute* ou *relative*.
- 13 Herdanza en CSS
- 14 Recursos externos

## Introdución

Có flotado e o posicionamiento CSS permite, en síntese, un máximo control sobre o lugar que cada elemento dunha páxina Web debe ocupar dentro da propia páxina, as súas condicións de visibilidade e "flotabilidade", así como o manexo de capas. Aquí veremos as propiedades CSS 2.1 para controlar o posicionamento dos elementos: *float*, *clear*, *position*, *bottom*, *top*, *left*, *right*, *overflow*, *clip*, *visibility* e *z-index*.

## Fluxo normal

Cando se fala de **fluxo normal** significa que os obxectos se dispoñen de arriba a abaixo e de esquerda a dereita. Flotando e posicionando con CSS conseguimos que os elementos deixen o fluxo normal.

# Flotar

Flotar serve para mover unha caixa á esquerda ou dereita ata que o seu borde exterior toque o borde da caixa que o contén ou outra caixa flotante.



Cando queremos flotar un elemento éste terá que ter definido implícita o expresamente o seu largo.

**As caixas flotantes non se atopan no "fluxo normal" do documento polo que as caixas que seguen o fluxo normal compórtanse coma se as flotantes non estivesen aí.**

Flotar é mais que mover unha imaxe, serve para crear deseños multicolumna, barras de navegación de listas non numeradas, aliñación en forma de táboa sen táboas, e moito máis. Para todo iso empregárase a propiedade *float*.

Cando colocamos unha imaxe nun documento e lle cambiamos a alineación con respecto ao texto, en realidade o que estamos facendo é flotar. Agora grazas a CSS podemos controlar todas esas propiedades.

## Caixas non flotantes

Para ver as consecuencias de flotar caixas facemos este exemplo base. Trátase de tres divs de distintas cores e tamaños, os tres están metidos no interior dun div chamado "contido" basicamente para que o conxunto se atope centrado no navegador.

### ◊ Arquivo html:

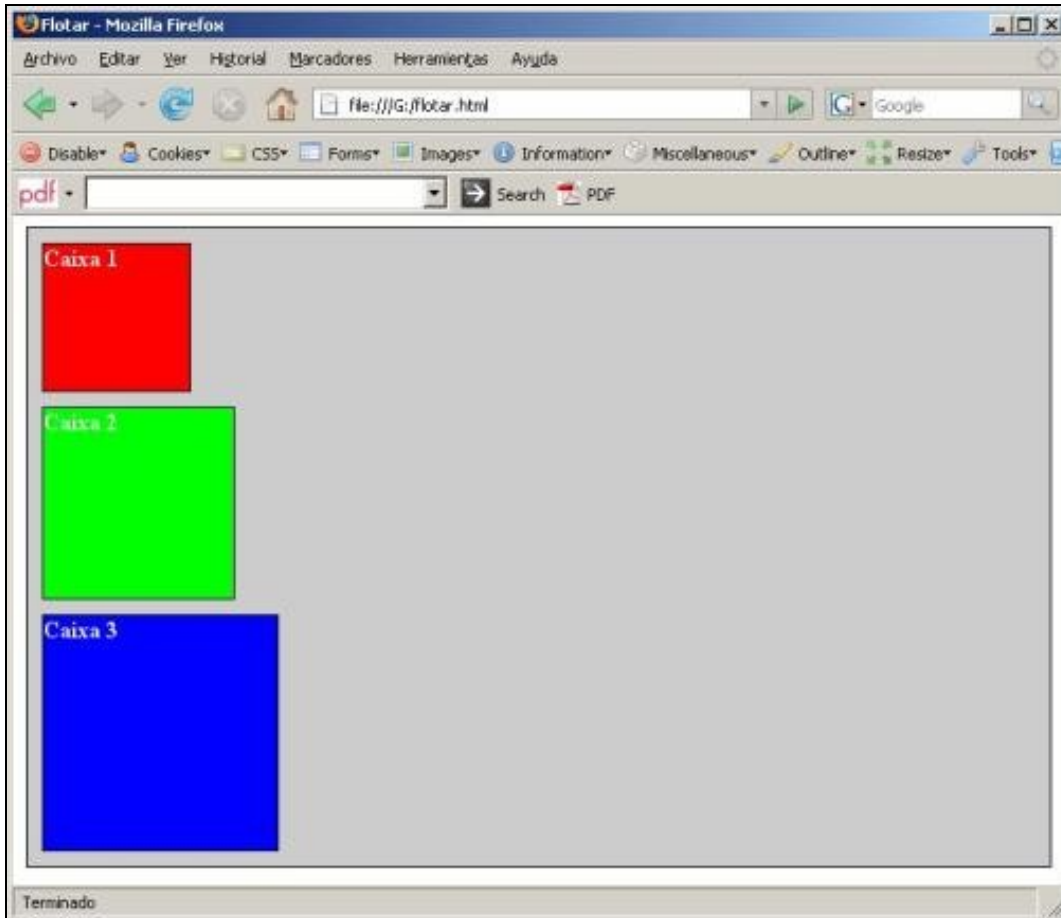
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <link rel="stylesheet" type="text/css" href="estilos.css" />
    <title>Flotar</title>
  </head>
  <body>
    <div id="contido">
      <div id="caixa1"> Caixa 1 </div>
      <div id="caixa2"> Caixa 2 </div>
      <div id="caixa3"> Caixa 3 </div>
    </div>
  </body>
</html>
```

### ◊ Arquivo estilos.css:

```
body {
  font-weight: bold;
  color: white;
}
#contido {
  /* Ancho de 700px e centrado no navegador */
  width: 700px;
  margin: auto;
  /* Borde sólido e fondo gris clariño */
  border: solid 1px black;
  background-color: #CCCCCC;
}
#caixa1 {
  width: 100px; height: 100px;
  background-color:#FF0000;
  border: solid 1px black;
  margin: 10px;
}
#caixa2 {
  width: 130px; height: 130px;
  background-color:#00FF00;
  border: solid 1px black;
```

```
margin: 10px;
}
#caixa3 {
width: 160px; height: 160px;
background-color:#0000FF;
border: solid 1px black;
margin: 10px;
}
```

Como se trata de tres elementos de bloque cada un deles comeza debaixo do anterior.

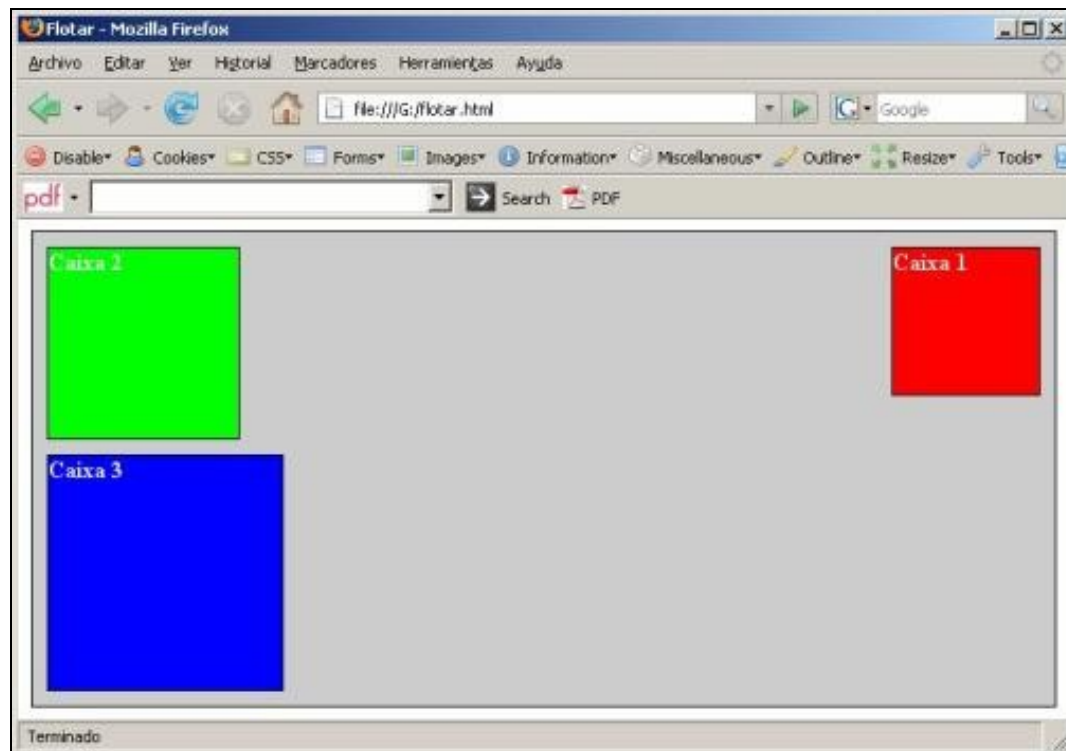


## Flotar unha caixa á dereita

Agora flotamos a Caixa1 á dereita, así que temos que modificar o selector #caixa1 do arquivo estilos.css, o resto deixáremolo igual:

```
#caixa1 {  
    width: 100px; height: 100px;  
    background-color:#FF0000;  
    border: solid 1px black;  
    margin: 10px;  
    /*Flotar á dereita */  
    float: right;  
}
```

Como se mostra na seguinte imaxe, cando se flota a Caixa1 á dereita, sácase do fluxo normal do documento e móvese á dereita ata que o seu borde dereito toca o borde dereito do bloque que o contén.



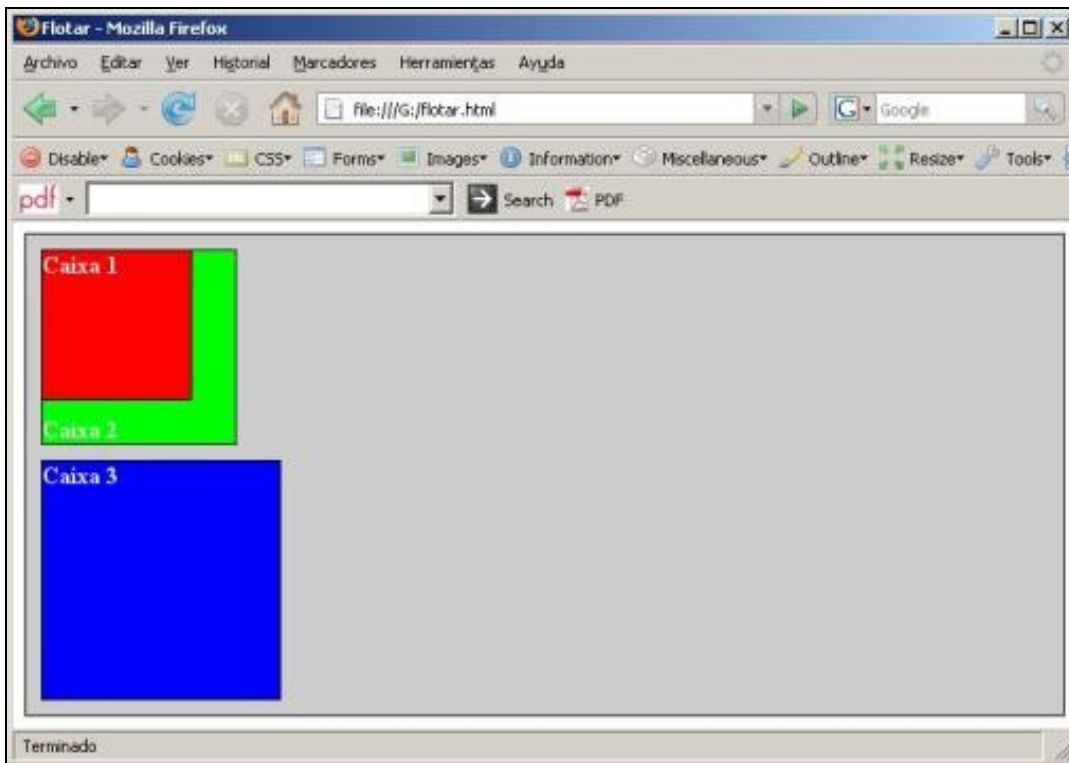
**Nota:** Recordade que non chega a "pegar de todo" pois no estilos.css configúrase para as tres caixas unha marxe de 10px en todos os lados.

## Flotar unha caixa á esquerda

Agora imos flotar a Caixa1 á esquerda, polo que se saca outra vez do fluxo do documento e móvese á esquerda ata que o borde esquerdo toca o borde esquerdo do bloque que o contén. Temos que modificar o selector #caixa1 do arquivo estilos.css, o resto deixarémolo igual:

```
#caixa1 {  
    width: 100px; height: 100px;  
    background-color:#FF0000;  
    border: solid 1px black;  
    margin: 10px;  
    /*Flotar á esquerda */  
    float: left;  
}
```

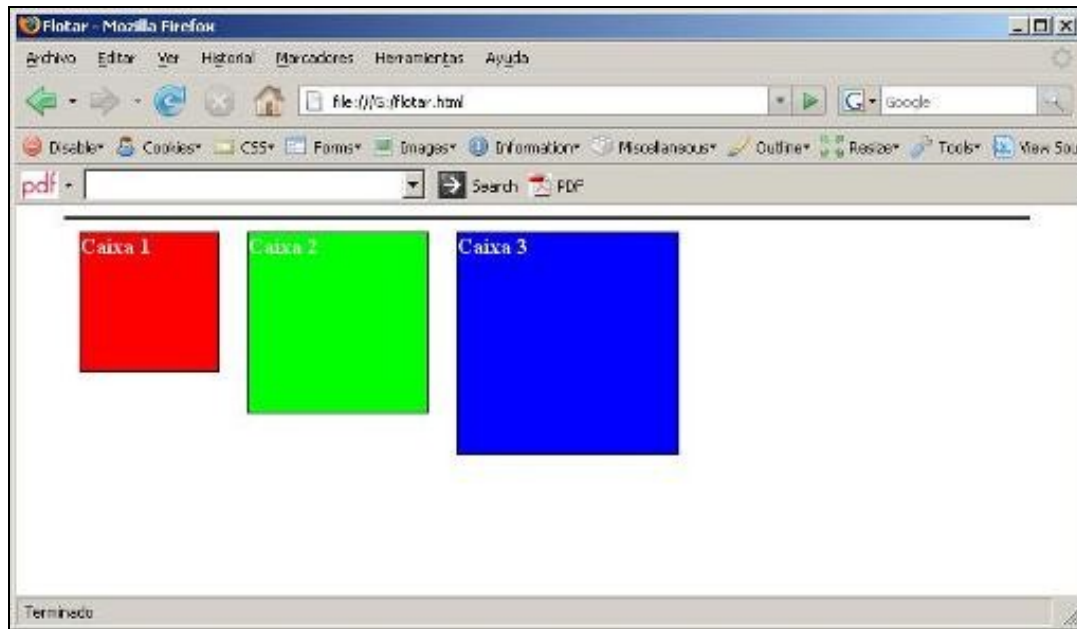
Posto que xa non se atopa no fluxo do documento, non ocupa espazo e sitúase sobre a Caixa2, ocultando así parte dela.



## Flotar todas as caixas á esquerda

Ó flotar as tres caixas á esquerda, Caixa1 móvese á esquerda ata que toca a caixa que o contén e as outras dúas caixas móvense á esquerda ata que tocan a caixa flotante anterior. Temos que modificar os selectores #caixa1, #caixa2 e #caixa3 do arquivo estilos.css tal e como se fixo no apartado anterior.

O resultado obtido podémolo ver na seguinte imaxe.



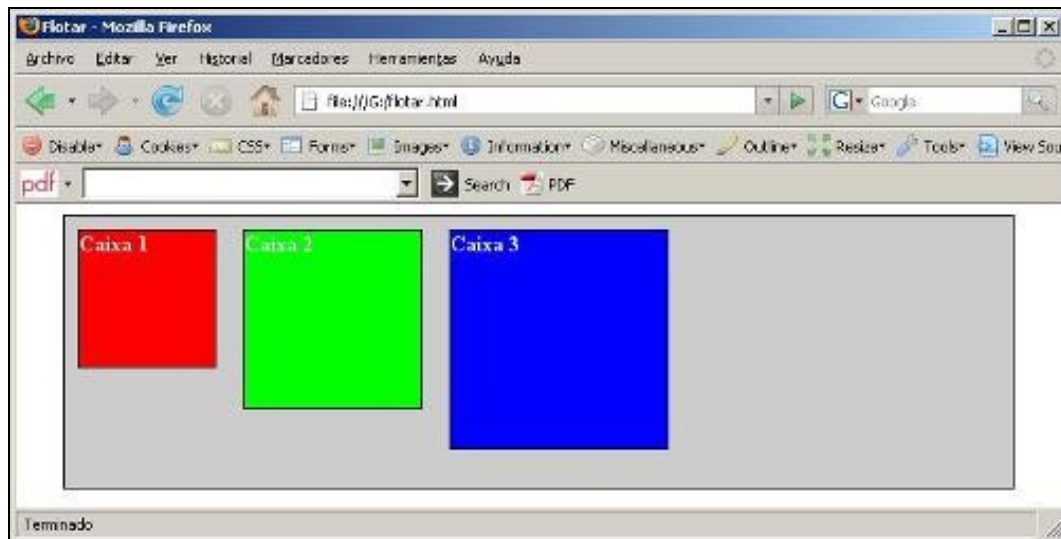
### Nota

Tal e como se ve na imaxe anterior, o div "contido" ten agora unha altura de 0px, xa que todo o que ten no seu interior está fóra do fluxo do documento. É dicir, non ten nada dentro e non precisa ter unha altura determinada.

Para que se volva a ver o "contedor":

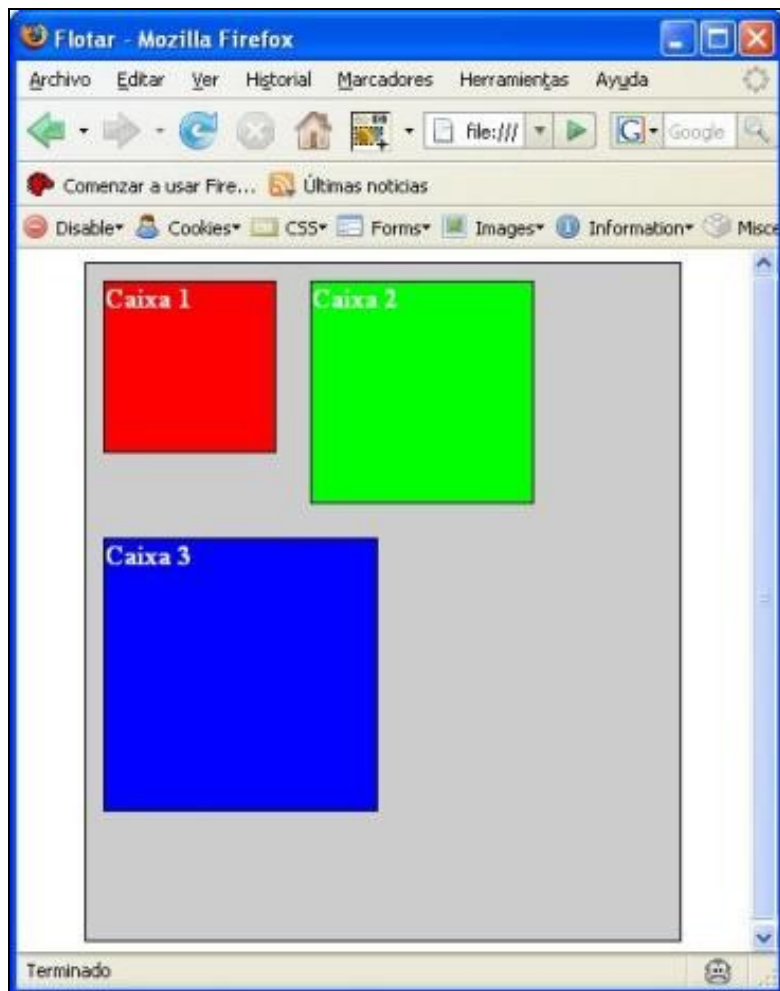
- ◇ Pódesele poñer unha altura configurando no selector #contido a propiedade *height* cunha altura maior que o alto da caixa máis grande.
  - Así e todo os divs de cores seguen estando no interior do div "contido" e iso vese en que non saen do ancho marcado por este e o aliñamento é con respecto a el.
- ◇ Tamén se pode flotar á esquerda a caixa con id #contido, así volvería a aparecer, iso si, flotada á esquerda.
- ◇ Engadir un div totalmente baleiro dentro do div "contido" e despois das tres caixas flotadas e que configuraremos coa propiedade **clear** có valor, por exemplo, **both**. Esta é, sen dúbida, a mellor solución.

A seguinte imaxe mostra o resultado de engadir ó selector #contido a propiedade *height: 200px*.

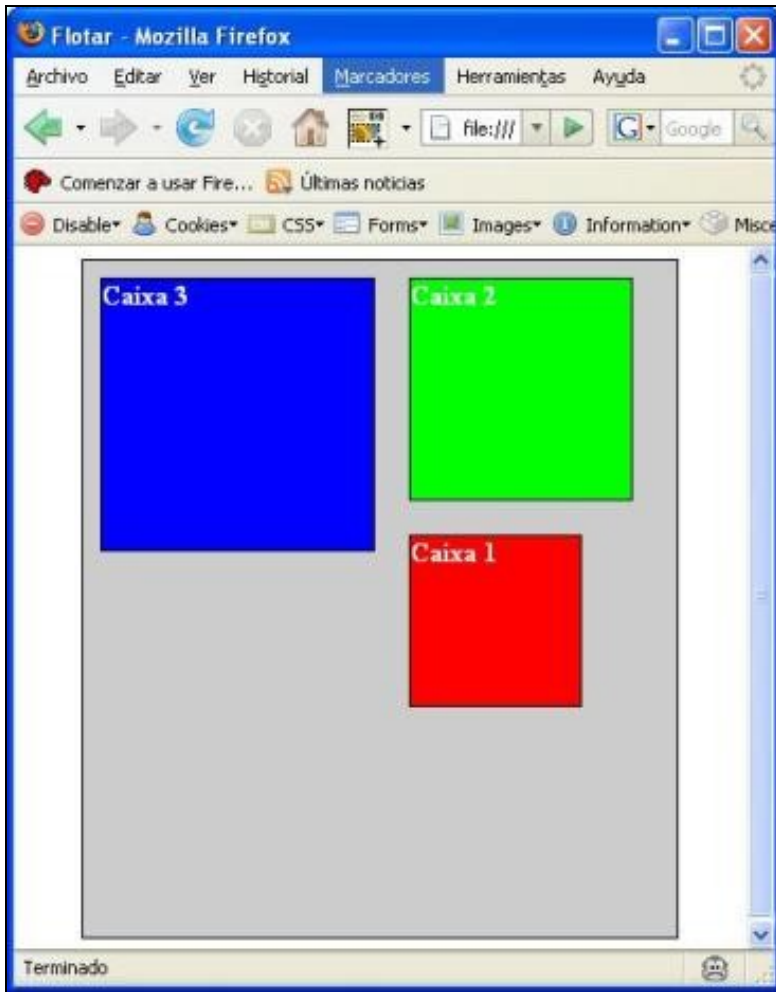


- **Bloque contedor demasiado estreito.**

Un caso curioso é cando o noso div "contido" é demasiado estreito como para que os elementos flotados collan horizontalmente. Se é así, o que non colla desprazarase cara abaixo. Podemos ver esta situación na seguinte imaxe, onde se configurou o ancho do div "contido" con un width de 350px (recordar que a suma dos anchos das tres caixas, mais as súas marxes, é maior que eses 350px).



A orde das caixas no documento HTML é Caixa1 - Caixa2 - Caixa3, pero se cambiamos a orde das caixas a Caixa 3 - Caixa2 - Caixa1 vemos na seguinte imaxe que o resultado cambia moito. Iso é debido a que a Caixa1 tropeza coa Caixa3 ó intentar ser flotada á esquerda.





## Marxes negativos e solapamento

As dúas grandes regras do flotamento é que os obxectos flotados nunca deberían ir mais alá da área de contido do bloque que os contén e que nunca debería solapar a outros elementos. Podemos romper estas pautas cando aplicamos marxes negativos:

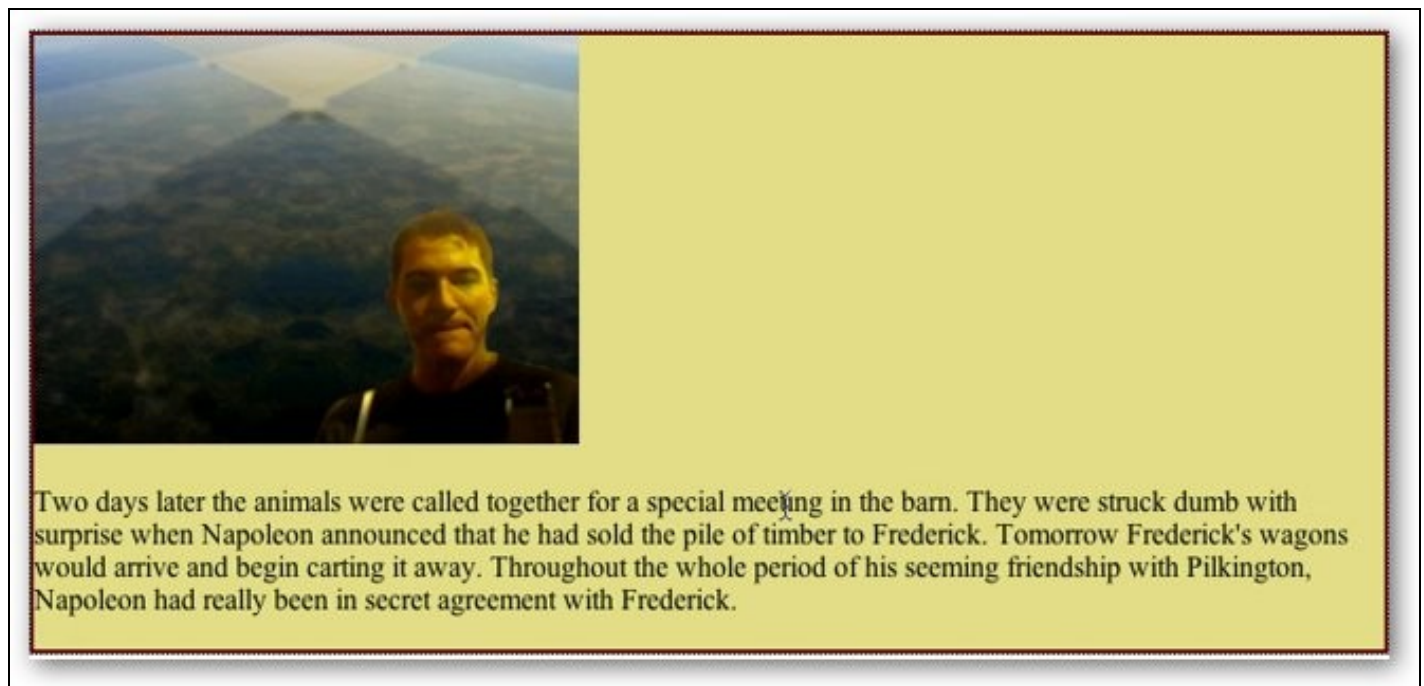
```
img {float: left; margin: -10px; }
```

## Limpar (*clear*)

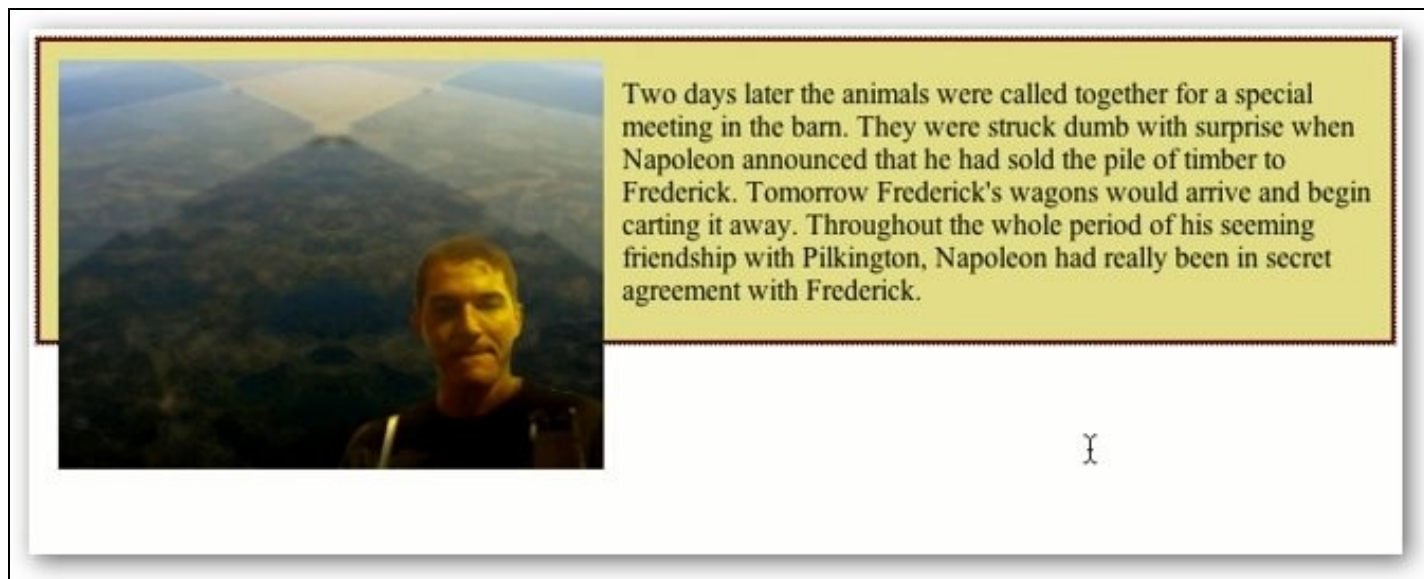
- Se se quere que a área que está ó lado do elemento flotado se manteña limpa e que o seguinte elemento comeze na súa posición normal no bloque que o contén, utilízase a propiedade *clear*.
- Pode ter varios valores: *left*, *right*, *both*, *none* e *inherit*.
- O valor *left* indica que o elemento comeza por debaixo de calquera elemento que fora flotado no extremo esquerdo do bloque ó que pertence.
- O valor *right* igual que *left* pero cambiando esquerda por dereita.
- O valor *both* move cara abaixo o elemento ata que estea limpo de elementos flotados a ambos lados.

Exemplo:

Partimos da seguinte imaxe e o seguinte texto:



Agora aplicámoslle á marca IMG un estilo no que lle indicamos que a imaxe flote a esquerda, e ademáis engadímoslle un padding de 10px. E obteremos o seguinte resultado:

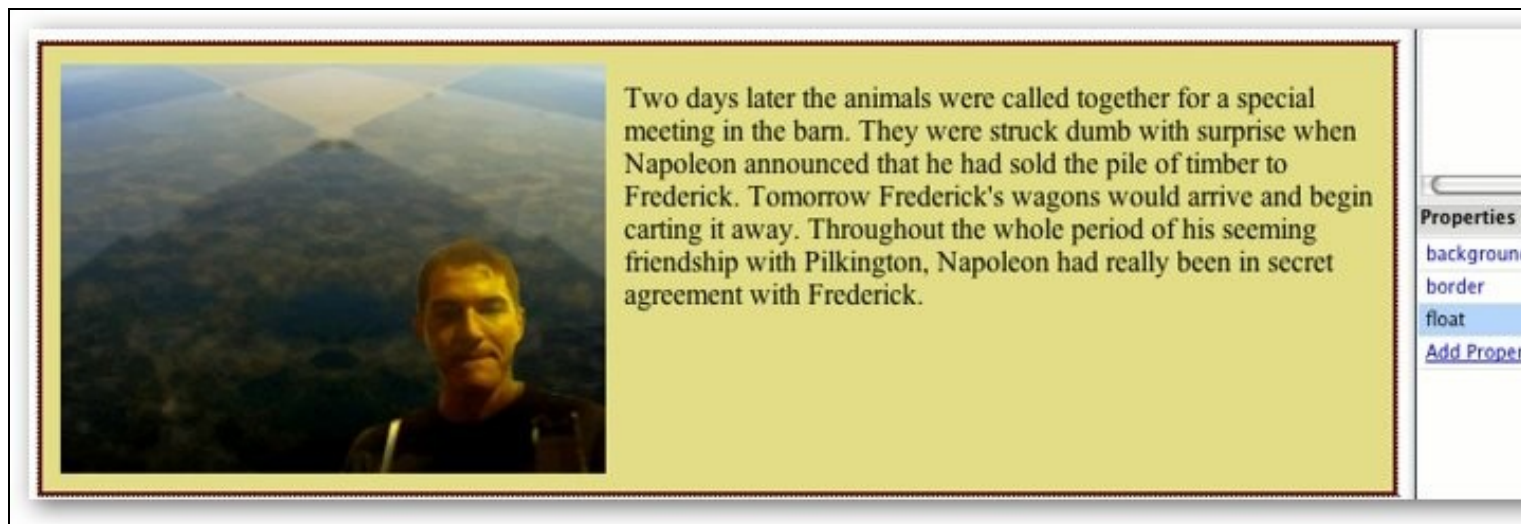


Podemos observar que a imaxe sigue no seu lugar (co padding de 10 px correspondente) e o texto sube para flotar ó arredor da imaxe.

O **problema** que atopamos é que o div principal non é suficientemente alto para rodeala imaxe, e isto é debido a que **cando flotamos unha imaxe ésta é sacada do fluxo normal do documento**. Polo tanto o div principal xa non ve a imaxe como parte do seu contido, polo que non pode rodeala.

Hai 2 formas de **solucionar este problema**:

**1ª.- Engadirlle ó contedor principal a propiedade float.** O facer isto o contedor principal abarcará o resto dos float anidados que teñamos dentro. O resultado é o seguinte:



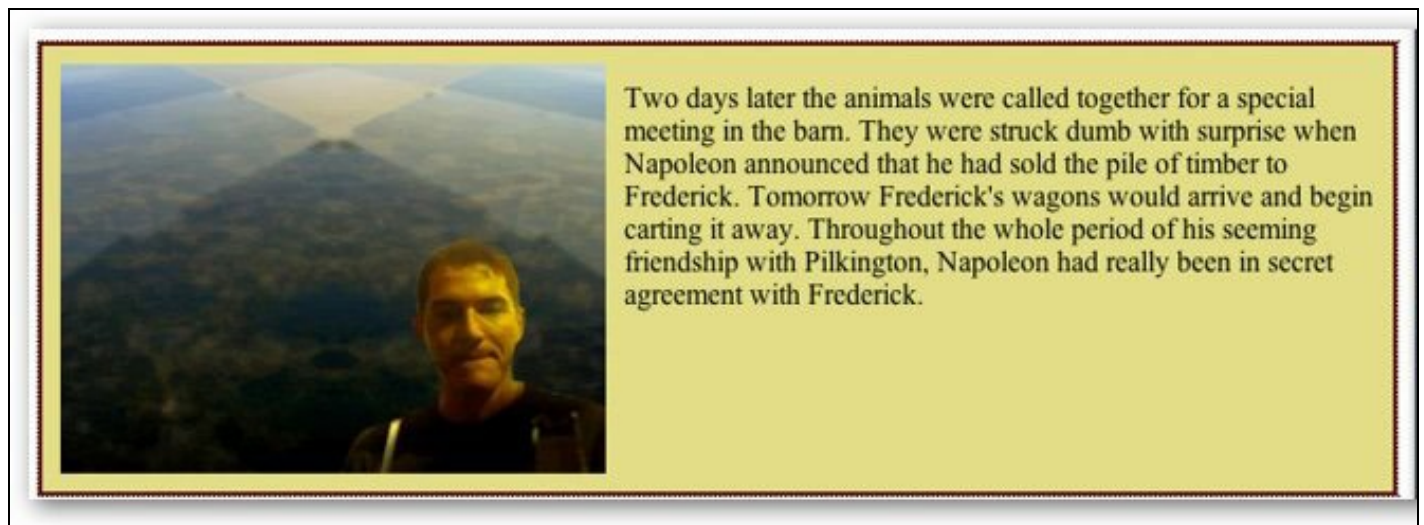
2ª.- Facer un clear both nalgún elemento que esté dentro do contedor principal (xeralmente o derradeiro elemento). Isto pódese facer engadindo por exemplo á marca BR dita propiedade clear.

Aquí vémo-lo código do exemplo, e observamos que temos o contedor principal, unha imaxe e un párrafo e ningunha marca máis. Polo que podemos engadir un BR ó termina-lo párrafo e o resultado sería o mesmo.

```
<body>
<div class="container">
  
  <p>Two days later the animals were called together for a
special meeting in the barn. They were struck dumb with
surprise when Napoleon announced that he had sold the pile of
timber to Frederick. Tomorrow Frederick's wagons would
arrive and begin carting it away. Throughout the whole period
of his seeming friendship with Pilkington, Napoleon had
really been in secret agreement with Frederick. </p>
</div>
</body>
</html>
```

Polo tanto cando temos un elemento cunha propiedade clear, que esté contido dentro de un outro, dita propiedade cancelará a partir de ahí os floats que estean activados e o contedor principal expandirase automaticamente ata abarcar tódolos contedor anidados ,incluídos os elementos con propiedade float.

E o resultado será polo tanto:



# Posicionamento

## Introdución ao posicionamento

Unha vez familiarizado có modelo de caixas é preciso pasar ós modelos de posicionamento e de formato visual para conseguir entender a forma na que se organizan os elementos nunha páxina.

## Propiedade *display*

Nun documento temos elementos de tipo bloque como *p*, *h1*, *div*,... e elementos "en liña" como *strong* e *span*. É posible cambiar o modo en que o documento interpreta estes elementos empregando a propiedade *display*.

- Para facer que un elemento "en liña" se comporte como un elemento de tipo bloque estableceremos a súa propiedade *display* en *block*.
- Para facer que un elemento non xere caixa estableceremos a súa propiedade *display* en *none*, isto fará que a caixa e todo o seu contido non se mostrará e non ocupará espazo no documento.

## Propiedade *position*

Para posicionar elementos nun documento temos a propiedade *position*. A propiedade ***position*** admite cinco valores: *absolute*, *relative*, *static*, *fixed* e *inherit*. Por defecto é *static* que será que segue o fluxo normal dado pola posición dese elemento no (X)HTML.

### • *position:static*

A posición *static* é o valor por defecto de todos os elementos HTML, e polo tanto non fai falla especificar este estilo. Os elementos en posición estática vanse debuxando un a continuación do outro, segundo as regras ordinarias, fluíndo coa páxina.

### • *position: relative*

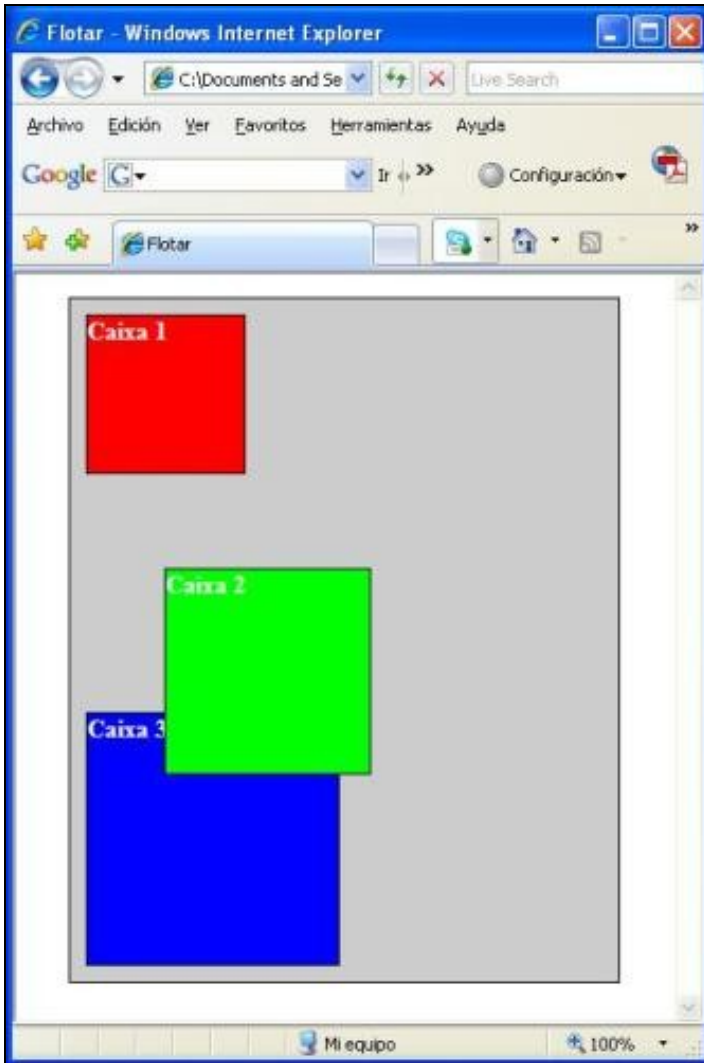
A posición relativa é un concepto moi sinxelo de entender. Se posiccionamos un elemento relativamente, permanecerá exactamente onde está. Logo, pode cambiar o elemento "relativo ao seu punto de partida" establecendo unha posición vertical ou horizontal. Podemos ver este exemplo no que se desprazou a Caixa2 50px do extremo esquerdo e 50 px do extremo superior da súa posición relativa.

**O CSS quedará así:**

```
body {
    font-weight: bold;
    color: white;
}
#contido {
    /* Ancho de 700px e centrado no navegador */
    width: 350px;
    margin: auto;
    /* Borde sólido e fondo gris clariño */
    border: solid 1px black;
    background-color: #CCCCCC;
}
#caixa1 {
    width: 100px; height: 100px;
    background-color:#FF0000;
    border: solid 1px black;
    margin: 10px;
}
#caixa2 {
    width: 130px; height: 130px;
    background-color:#00FF00;
    border: solid 1px black;
    margin: 10px;
    /* Posicionamento relativo */
    position: relative;
    left: 50px; top: 50px;
}
#caixa3 {
    width: 160px; height: 160px;
    background-color:#0000FF;
    border: solid 1px black;
```

```
margin: 10px;  
}
```

E obteremos o seguinte resultado:

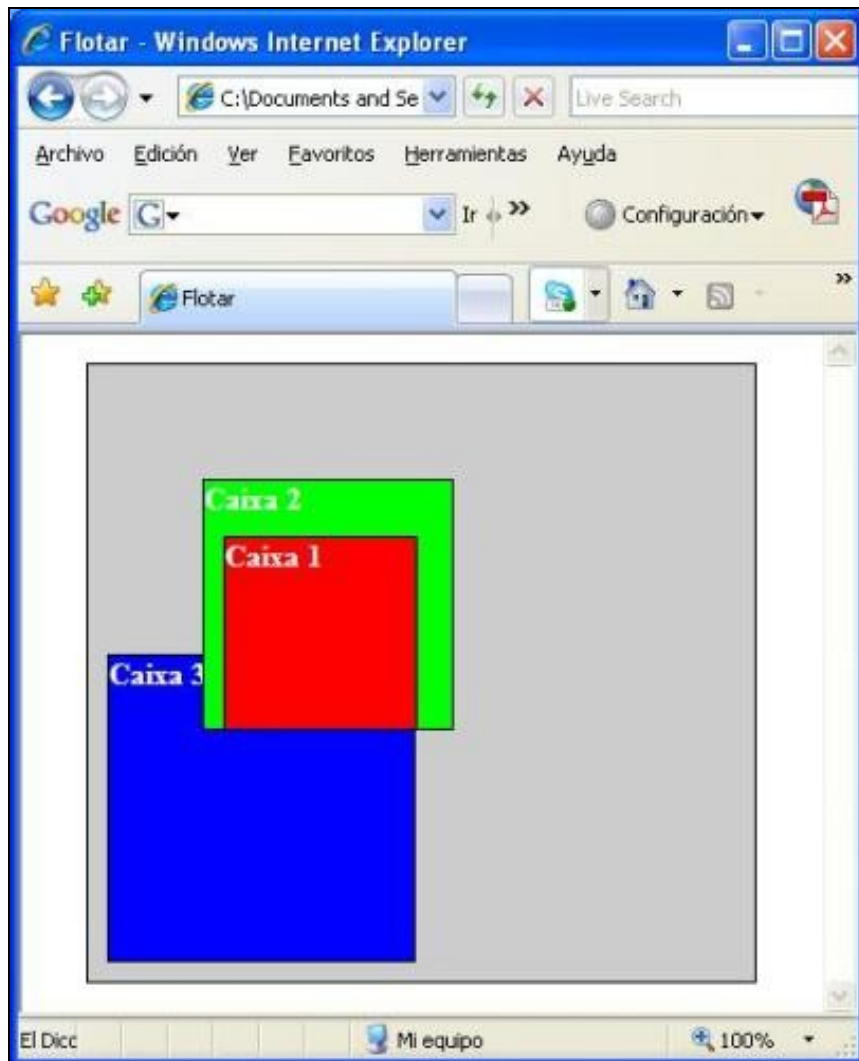


Fixádevos como se despraza a caixa verde pero o resto do documento segue supoñendo que se atopa na súa posición orixinal (como se non se movera).

Outro elemento posicionado relativamente que lle siga inmediatamente no fluxo HTML calculará a súa orixe como segue:

1. Se o seguinte elemento é un fillo do anterior, a orixe deste seguinte estará no final do anterior.
2. Se o seguinte elemento non é un fillo do primeiro elemento, este seguinte elemento terá a súa orixe onde o primeiro elemento tería o seu final se non se fixaran as súas propiedades *top* e *left*.

Para demostrar este segundo punto, mira como queda ó modificar o html e meter a Caixa 1 dentro da Caixa 2 que segue posicionada de xeito relativo:



- **position: absolute**

A posición absoluta saca o elemento do fluxo do documento, non ocupando ningún espazo. Outros elementos no fluxo normal do documento actuarán como se o elemento posicionado absolutamente nunca estivese aí.

Se o noso elemento absoluto non está contido dentro de ningún outro, a posición de coordenadas mídese respecto do inicio da etiqueta BODY (contedor principal). Se o o noso elemento absoluto está contido dentro doutro elemento, e este contedor está posicionado absoluta ou relativamente, as coordenadas do elemento absoluto "contido" calcúlanse dende a orixe do elemento contedor.



## Determinar orixe coordenadas

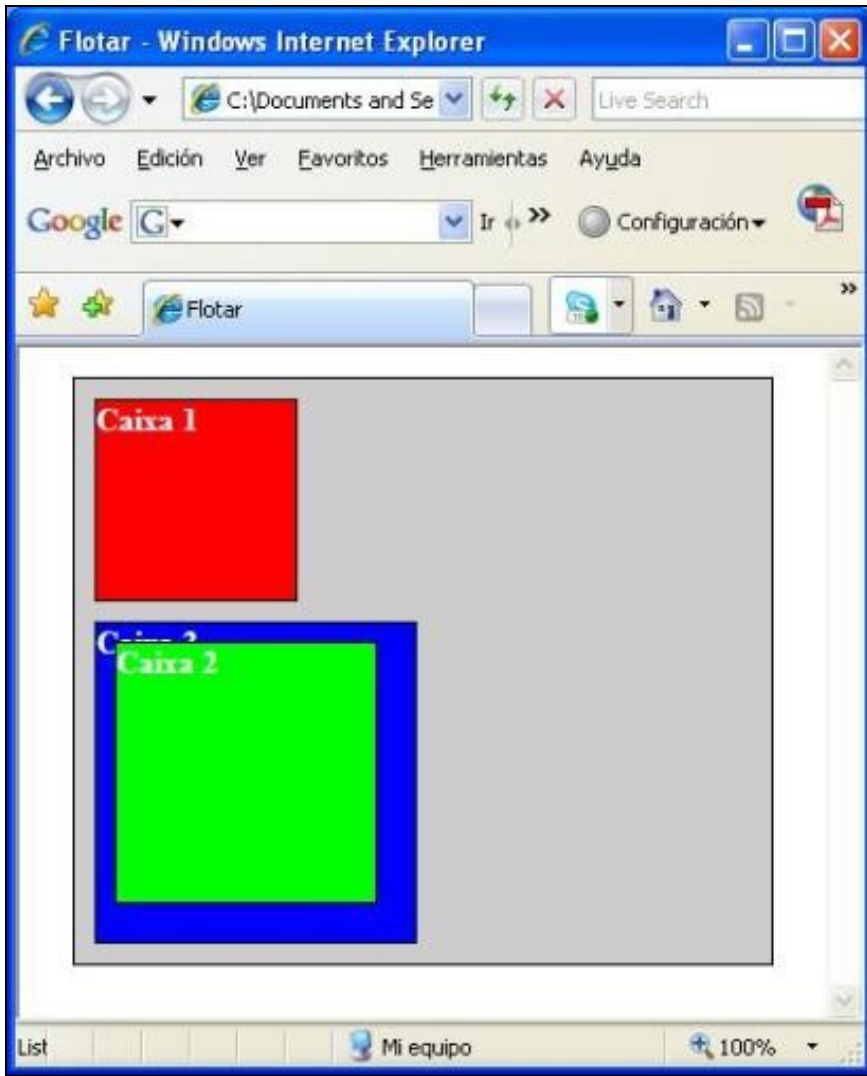
Determinar a orixe de coordenadas a partir do cal se despraza unha caixa situada de forma absoluta é un proceso complexo que se compón dos seguintes pasos:

- Búscanse todos os elementos colectores da caixa até chegar ao elemento da páxina.
- Percórrense todos os elementos colectores empezando polo máis próximo á caixa e chegando até o BODY
- De todos eles, o navegador queda co primeiro elemento colector que estea situado de calquera forma diferente a `position: static`
- A esquina superior esquerda dese elemento colector situado é a orixe de coordenadas.

Os elementos "absolutamente" posicionados dentro dun documento HTML (e tamén os elementos *children* que engloba) non forman parte do fluír xeral do resto de elementos do documento. A súa posición é asignada independentemente da que ocupen o resto dos elementos do documento.

Configúrase a posición coas propiedades *top* e *left*: coa propiedade **top** indicamos a posición vertical da esquina superior esquerda do elemento a posicionar con respecto ao elemento anterior existente na páxina e, coa propiedade **left** a posición horizontal desa esquina con respecto ao elemento contedor.

Se configuramos a propiedade *position* da Caixa 2 como *absolute* o resultado é o seguinte:





- **position: fixed**

- ◆ O posicionamento fixo é como o posicionamento absoluto pero o elemento posicionase esta vez en relación o porto de visualización (normalmente, a fiestra do navegador).

### Exemplos de posicionamento absoluto

## Propiedade *visibility*

A propiedade *visibility* dun elemento controla se este será visualizado ou non, segun a fixemos en **visible** ou **hidden**.

**Aínda que o elemento non sexa visible, continúa ocupando o seu espazo no fluír do documento.** É dicir, ao contrario do que ocorría coa propiedade *display* (incluída dentro das propiedades de Clasificación): se a propiedade *display* fíxase en *none*, o espazo que debería ocupar este elemento é reutilizado.

## Propiedade *clip*

A propiedade *clip* permite decidir que parte dun elemento será visible.



**Nota**

Non afecta o espazo ocupado polo elemento no fluxo HTML.

O exemplo mostra a sintaxe para determinar a zona visible, usando regras de estilo in line, ou scripts.

A orde dos parámetros ha de ser: top, right, bottom, left:



**Nota**

Para que o *clip* funcione a posición debe axustarse a *absolute*.

```
<IMG ID=Olivial STYLE="position:absolute; top:0px; left:0px; width:136px; height:228px; clip:rect(0 136 228 0)" SRC=olivial.gif>
```

Nesta dirección podedes atopar un uso interesante da [función clip para crear thumbnails](#).

## Propiedade *overflow*

A propiedade *overflow* permite elixir o que sucede co contido dun elemento cando este non encaixa ou coincide co rectángulo definido por algunha das propiedades *top*, *left*, *height* e *width*. Son catro posibles valores para esta propiedade:

1. **Visible** -- Expande o tamaño do elemento até dar encaixe a todo o contido, é o valor por defecto.
2. **Hidden** -- Axusta o tamaño do contido ao declarado do elemento.
3. **Auto** -- Coloca barras de *scroll* no elemento só se o seu contido excede do tamaño fixado.
4. **Scroll** -- Coloca barras de *scroll* independentemente do tamaño do elemento.

Exemplo de *overflow*:

```
<DIV STYLE="position:absolute; top:60px; left:0px; width:100px; height:100px; overflow:visible; background-color:red; ">
Isto é corto
</DIV>

<DIV STYLE="position:absolute; top:60px; left:200px; width:100px; height:100px;
```

```
overflow:scroll; background-color:green; ">
Isto é largoooooooooooo
</DIV>
```

## Propiedade *z-index*

Se cando definimos algun elemento con posición absoluta, este ten que visualizarse no mesmo lugar ocupado por outro elemento, producirase unha superposición de elementos visualizándose, na parte coincidente, só o que está ocupando a posición ou capa superior.

Por defecto, os elementos se apilan na orde en que aparecen: o elemento situado máis tarde no arquivo orixe quedará arriba.

A propiedade **z-index** permite especificar o **z-orde dos elementos**, isto é, a orde de apilamiento en capas do documento.



### Nota

Os elementos con maior Z-INDEX son colocados encima dos que teñan menor Z-INDEX , quedando estes últimos tapados polos primeiros. A propiedade Z-INDEX só se aplica a elementos que teñan a propiedade position en absolute ou relative.

Exemplos de overflow e z-index.

Tres divs de 100 \* 100 px, con diferentes overflows

overflow:visible

123456789101112131415

overflow:hidden

1234567891011

overflow:auto

1234567891011



Tres divs con diferente z-index.

z-index=1

z-index=2

z-index=3

z-index=1

z-index=3

z-index=2

Neste exemplo o bloque verde é fillo do bloque gris, e aínda que ten un z-index de 3 aparecerá por debaixo do bloque vermello, xa que o cálculo do z-index é feito entre bloques do mesmo nivel.

## Herdanza en CSS

CSS traballa na herdanza cunha valor denominado **inherit**. Cando se aplica a unha propiedade dun elemento este valor, dito elemento terá o mesmo valor que ten o elemento pai.

Por exemplo, con esta folla de estilos:

```
.un{
  background-color: blue;
  color: black;
}

.dous{
  background-color: inherit;
  color: inherit;
  font-weight: normal;
}
```

E o seguinte fragmento HTML

```
<div class="un">
  <p class="dous">
    Hola mundo, isto é un párrafo !
  </p>
</div>
```

A cor de fondo do elemento div "un" é de cor azul, porque a cor de fondo da propiedade está establecida en azul.

A cor de fondo do parágrafo é tamén de cor azul, porque a cor de fondo está establecida como inherit, e polo tanto herdará do elemento pai o seu color de fondo e tamén o color do texto por que tamén ten a propiedade como inherit.

Hai que ter en conta que non tódalas propiedades poden ser herdadas.

## Recursos externos

- **Visita o seguinte [Recurso externo](#), onde poderás atopar uns exemplos moi bos sobre *Posicionamento*.**
- [Exemplo posicionamento](#)
- [Posicionamento e Visualización](#)

--Vieites e Veiga 10 Nov 2008.