

# Posicionament

# Introducció

Un document HTML és un document **viu**

Fins i tot sense aplicar cap CSS, un document HTML ja té les seves pròpies regles:

- **fluidesa**: com s'adapta el contingut a les dimensions del navegador
- **ordenació**: en què apareixen els elements d'ordre
- **apilament**: com apareixen els elements uns sobre els altres

# Fluidesa

En HTML, el **contingut** és el més important.

Tots els elements tipus *bloc* són **fluids**. Naturalment, adaptaran el seu disseny per adaptar-se al seu contingut interior:

- **amplada: 100%:** un bloc ocuparà tota l'amplada disponible
- **l'ajust de línia:** si el contingut en línia d'un bloc no encaixa en una sola línia, continuarà en una línia nova
- **altura: automàtica:** l'alçada d'un bloc varia automàticament per coincidir amb la mida del seu contingut

# Ordenació

Els elements HTML es mostren en l'**ordre** en què s'escriuen **al codi**.

Primer al codi -> primer al navegador.

Cada etiqueta de tipus *block* apareix en l'ordre en què apareixen al codi HTML, de dalt a baix.

```
<div>Primer</div>  
<div>Segon</div>  
<div>Tercer</div>  
<div>Quart</p>  
<div>Cinquè</div>
```

# Apilament

L'**ordre de pila** depèn de com estiguin uns **dins** dels altres: els elements fills apareixen **a sobre** dels seus respectius pares.

```
<div class="pare-i-fill">  
  Aquest pare està enrere  
  <p>  
    Aquest paràgraf apareix <strong>a sobre</strong> del seu pare  
  </p>  
</div>
```

# Trencant el flux

Tot i que el comportament predeterminat del navegador és *eficient*, pot ser que no sigui *suficient* per a les vostres necessitats de disseny.

Diverses propietats CSS permeten **trencar** el flux:

- `width` i `height` poden alterar la **fluidesa** d'un element
- `position` amb valors com `aboslut` o `fixed` **elimina** un element del flux
- `float` **canvia** el comportament d'un element així com el seu entorn
- `z-index` pot alterar l'ordre en què els elements s'apilen

## position: static

És el valor per defecte. Ubica la caixa a la part esquerra superior.

```
/*  
S'aplica a tots els div de la classe test-static.  
Si no posés res, tindria el mateix comportament.  
*/  
div.test-static {  
    position: static;  
}
```

## position: relative

És com el posicionament estàtic, però permet fer un desplaçament. El desplaçament s'especifica amb les propietats `top`, `right`, `bottom` i `left`.

```
/*  
S'aplica a tots els div de la classe test-relative.  
Ubica el div a la posició que tindria per defecte  
però desplaçant-lo 40 pixels cap a la dreta, és a dir,  
40 pixels des de l'esquerra.  
*/  
div.test-relative {  
    position: relative;  
    left: 40px;  
}
```



## position: absolute

Serveix per posicionar la caixa respecte al navegador. La posició s'especifica amb les propietats `top`, `right`, `bottom` i `left`.

```
/*  
Ubica el div 300 píxels per sota la part superior  
i 50 píxels a la dreta.  
*/  
div.test-absolute {  
    position: absolute;  
    top: 300px;  
    right: 50px;  
}
```

## position: fixed

És com el posicionament absolut però sense desplaçament de l'scroll. La posició s'especifica amb les propietats `top`, `right`, `bottom` i `left`.

```
/*  
Ubica el div a la part inferior esquerra,  
separat 200 píxels de la part inferior.  
A més, l'scroll no el desplaçarà  
*/  
div.text-fixed {  
    position: fixed;  
    bottom: 200px;  
}
```

# float

La propietat float desplaça les caixes a l'esquerra o dreta (left / right). El posicionament float es basa en ocupar l'espai disponible a esquerra o dreta.

```
/*  
S'ubiquen els div's float-r i float-l a l'esquerra i la dreta,  
sempre que hi càpiguen. Si no, un a sobre de l'altre  
*/  
div#float-l {  
    float: left;  
    width: 200px;  
}  
div#float-r {  
    float: right;  
    width: 200px;  
}
```

# clear

La propietat clear es combina amb la propietat float per a controlar fins a quin punt volem que es desplacin les caixes. Per exemple, si es fa servir float en un caixa (un div) i es vol ubicar una caixa a sota es fa servir també la propietat clear.

```
/*  
Ubica el div a la dreta i no en té  
cap d'altre flotant a cap de les dues bandes  
*/  
div#test-clear {  
    float: right;  
    clear: both;  
}
```

## On farem servir float i clear?

Les propietats `float` i `clear` tan sols les farem servir per a mostrar imatges integrades amb el text.

```
p.text-amb-imatge {  
    clear: both;  
}  
p.text-amb-imatge img {  
    float: left;  
    margin-right: 5px;  
}
```

Es pot fer servir també per fer posicionament més avançat, però hi ha alternatives més modernes i fàcils.