

# Mises en page en colonnes sans tableau

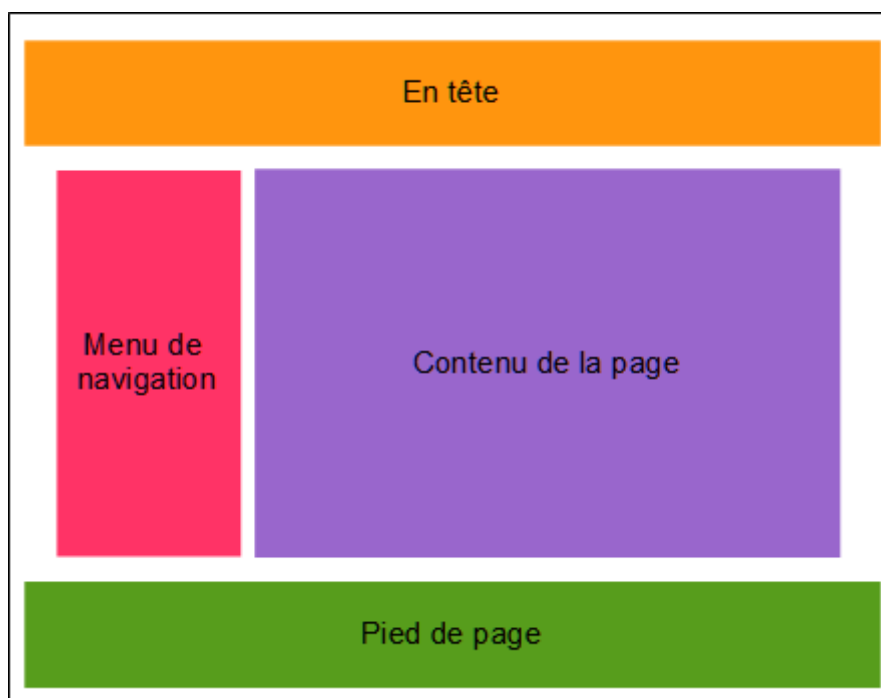
## Code (x)html

Le principe est très simple : on divise chaque élément de la page par un `div`. La page se divise en quatre sections, chacune correspondant à une zone précise de la page :

- Un en-tête : `div id="entete"`
- Un menu : `div id="menu"`
- Le contenu : `div id="contenu"`
- Un pied de page : `div id="footer"`

Pour faciliter la suite de la mise en forme, les boîtes `menu` et `contenu` seront englobées dans un autre cadre `main`.

Le but de l'opération est donc de placer les cadres `menu` et `contenu` l'un à côté de l'autre.



```
<div id="entete">  
  En tête  
</div>
```

```
<div id="main">  
  <div id="menu">  
    Menu  
  </div>
```

```
<div id="contenu">  
  Contenu  
</div>
```

```
</div>
```

```
<div id="footer">  
  Pied de Page  
</div>
```

## 1)En positionnement Flottant

Entre `menu` et `contenu`, celui à déclarer en flottant doit être `menu`, car il est le **premier des deux dans le flux** (le code html lu de haut en bas).

Pour éviter le problème de fusion de marges qui empêche les différents blocs de "coller" entre-eux, on attribuera à chacun d'eux une marge intérieure haute et basse de 1 pixel (`padding:1px 0`).

Si on désire un menu de 240 pixels de large et que le contenu ne se "rabatte" pas vers le bord gauche sitôt la hauteur du menu dépassée, on lui attribuera une marge gauche d'autant ou d'un peu plus.

Pour ne pas que le pied de page `footer` soit concerné par ce phénomène de flottaison (ce qui risque de le faire anormalement remonter), il faudra lui attribuer un `clear` qui remet à zéro la sortie du flux.

Pour finir, les écrans ayant tendance à devenir de plus en plus grands et une zone de lecture étant pénible à lire si elle trop large, on peut aussi attribuer au `main` une largeur maximum à ne pas dépasser, 960 pixels par exemple, que l'on peut centrer à l'aide de `margin:auto`.

On peut aussi positionner le menu à droite sans toucher au code html ! Ce qui est bien sûr impossible avec une mise en page par tableaux. Cela dit, la logique du flux n'est plus respectée (ce qui est placé en premier dans le code se trouve après un élément positionné à gauche de l'écran), ce qui est déconseillé par les normes accessiweb.

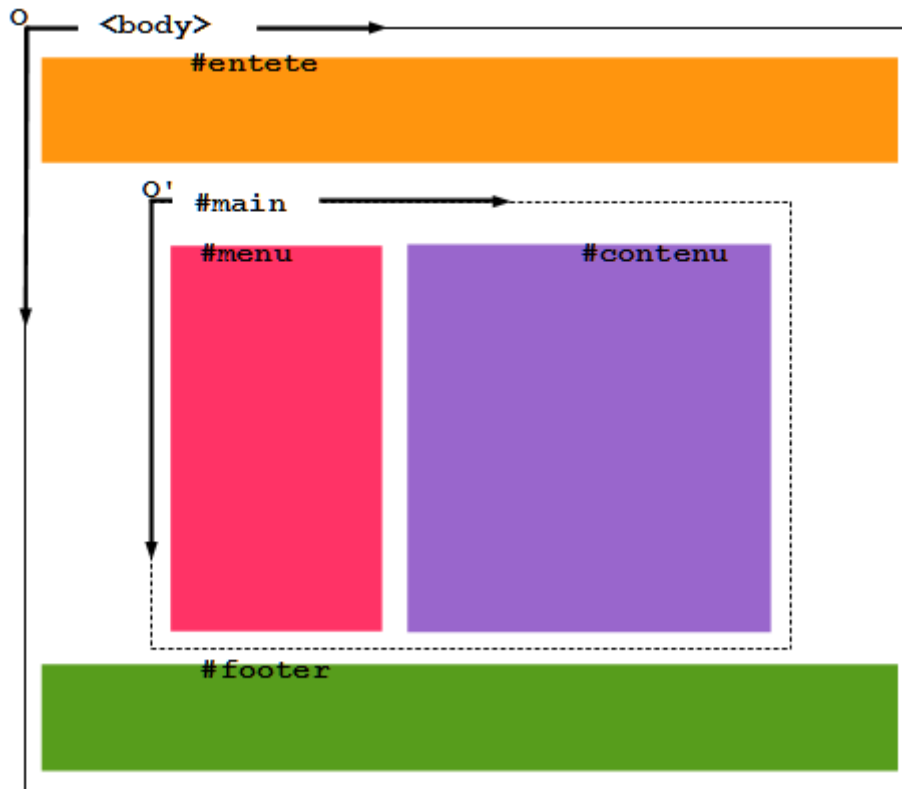
```
#menu {  
  float:right;  
  width:240px;  
}  
#contenu {  
  margin-right:245px;  
}
```

## 2)En positionnement absolu

Un élément positionné en `position:absolute` sort totalement du flux et est donc sujet à provoquer des superpositions. Pour cela, il sera donc plus prudent de déclarer une hauteur au bloc non positionné. Ne pouvant pas prévoir la hauteur que prendra le contenu, il vaut mieux, dans notre exemple, positionner le menu.

L'origine du repère à partir duquel on peut positionner l'élément, à l'aide des propriétés `top`, `bottom`, `left` ou `right`, se fait à partir du dernier ancêtre positionné, ou, à défaut, du coin haut à gauche de la fenêtre.

En d'autres termes, par défaut l'origine O est en haut à gauche. Mais on peut effectuer un changement de repère en O' à l'aide d'un élément parent aux éléments à positionner. Ici le `main` (bloc parent `enposition:relative`) va permettre de changer de repère pour `menu` et/ou `contenu` (éléments enfants).



Ce changement de repère va permettre d'aligner verticalement les deux blocs.

## Conclure