

# Div et CSS : une mise en page rapide et facile

par [Pierre-Baptiste Naigeon](#)

Date de publication : 19 Avril 2006

Dernière mise à jour : 04 Mai 2006

La mise en page en utilisant des tableaux est fortement déconseillée. Mais comment faire autrement ? Cet article va vous expliquer pas à pas comment arriver aux mêmes résultats, le plus simplement du monde.

I - Page simple : bandeau, contenu, pied de page.

II - Ajout d'une barre de menu à gauche du contenu.

III - Je voudrai séparer verticalement mon menu en deux blocs égaux... Je peux ?

IV - Et un encart de news en haut à droite de mon contenu, sans que ca prenne toute la hauteur, c'est possible ?

Tout au long de cet exercice, nous allons utiliser deux fichiers :

- index.html : contiendra tout notre code HTML.
- style\_div.css : contiendra toutes les informations relatives au positionnement de nos différents blocs.

Pour inclure notre CSS dans notre page HTML, rien de plus simple, il nous suffit d'insérer le code suivant entre les balises <head> et </head> :

```
<link rel="stylesheet" type="text/css" href="style_div.css">
```

Définissons tout d'abord les propriétés de base de nos blocs. Dans notre fichier CSS :

```
div {  
    text-align:center;  
}
```

Ce petit bout de code définit simplement que pour tous les éléments de type div, le texte sera centré ([text-align](#)).

## I - Page simple : bandeau, contenu, pied de page.

Nous allons commencer par créer un modèle de page tout bête, comprenant un bandeau, une zone de contenu, et un pied de page

Nous allons définir (parfaitement arbitrairement) des couleurs (uniquement pour bien se représenter les positions) ainsi que les dimensions de nos différents blocs :

- Bandeau : 600 x 50 px - couleur : #00CCFF
- Contenu : 600 x 400 px - couleur : #FFCC00
- Pied de page : 600 x 50 px - couleur : #33FF99

Ce qui va se transcrire dans notre fichier CSS par :

```
div#bandeau {  
    width:600px;  
    height:50px;  
    background-color:#00CCFF;  
}  
div#contenu {  
    width:600px;  
    height:400px;  
    background-color:#FFCC00;  
}  
div#piedpage {  
    width:600px;  
    height:50px;  
    background-color:#33FF99;  
}
```

Vous remarquerez le signe '#' placé après 'div', et suivi d'un descriptif. Il va permettre de définir un identifiant (unique), auquel s'appliquera le style.

Ici, nous avons donc défini trois identifiants : 'bandeau', 'contenu', et 'piedpage'. Les noms sont, je pense, assez explicites pour ne pas avoir besoin d'en rajouter ;)

Pour chacun des identifiants, nous avons défini trois propriétés :

- [width](#) : la largeur de notre bloc
- [height](#) : la hauteur du bloc
- [background-color](#) : la couleur de fond du bloc

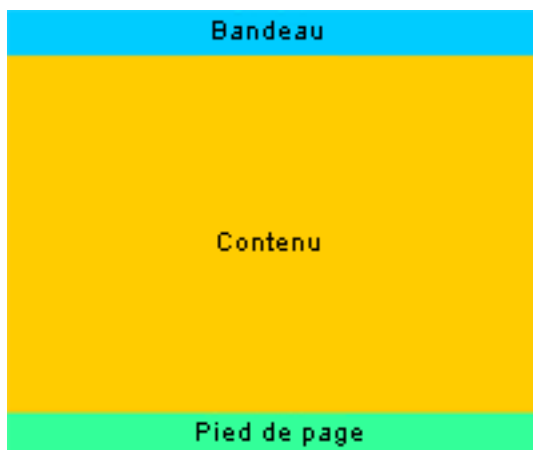
Certains éléments sont redondants, mais laissons les tels quels, ils ne le seront plus dans les exemples suivants.

Une fois les propriétés de ces trois éléments définies, il ne nous reste plus qu'à demander à notre HTML de nous les afficher.

Le code suivant est à placer entre les balises <body> et </body> de notre page :

```
<div id="bandeau">Ceci est le bandeau</div>
<div id="contenu">Ceci est le contenu</div>
<div id="piedpage">Ceci est le pied de page</div>
```

Ouvrez maintenant votre page index.html à l'aide de votre navigateur préféré, et ohhhh, magie, vous apercevez vos trois éléments positionnés les uns au dessus des autres :



*Résultat de l'exemple*



[Télécharger les codes-source de l'exemple](#)

## II - Ajout d'une barre de menu à gauche du contenu.

C'est bien joli tout ça, mais des sites sans menus, il n'y en a plus des masses. Qu'à cela ne tienne, nous allons en rajouter un...

Vous aurez sans doute remarqué dans l'exemple précédent que, lorsqu'on met dans notre fichier HTML des divs les uns à la suite des autres, ils s'affichent les uns sous les autres.

Pour afficher deux div l'un à côté de l'autre, il va donc nous falloir utiliser une autre propriété. C'est la propriété [float](#) qui va s'en charger.

Nous allons commencer par rajouter les propriétés du nouveau bloc de menus que nous allons créer :

```
div#menu {  
    float:left;  
    width:100px;  
    height:400px;  
    background-color:#FF6699;  
}
```

Il nous faut également rajouter la propriété [float](#) à l'élément 'contenu', sinon celui-ci va se 'glisser' sous le menu :

```
float:left;
```

Intéressons nous maintenant à notre menu dans la page HTML entre le bandeau et le contenu :

```
<div id="menu">Ceci est le menu</div>
```

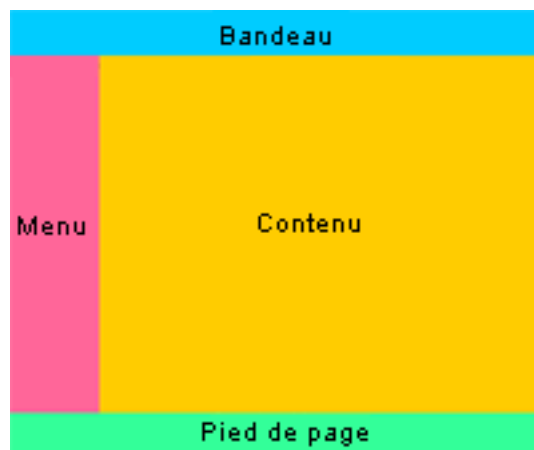
Mais, mais, mais !!! Qu'est devenu notre beau pied de page ? Et bien il s'est tout simplement glissé sous le menu et le contenu. Vous pouvez le vérifier simplement en modifiant la largeur de celui-ci, et vous verrez alors apparaître un petit bout de barre vert qui dépasse.

Pour éviter ce fâcheux problème, il existe là encore une propriété qui va permettre de dire qu'on veut que notre élément se remette à se comporter normalement (qu'il se positionne en dessous du reste quoi !). C'est la propriété [clear](#).

Rajoutons donc cette propriété à notre bloc 'pied de page' :

```
clear:both;
```

Et hop, c'est encore magique, nous obtenons le résultat escompté :



*Ajout d'un menu*



[Télécharger les codes-source de l'exemple](#)

### III - Je voudrai séparer verticalement mon menu en deux blocs égaux... Je peux ?

Décidément, vous êtes bien contrariants ;) Mais c'est possible, une fois qu'on a bien assimilé les principes ci-dessus.

Petit rappel : sans propriétés spécifiques, les blocs se positionnent les uns sous les autres. En utilisant la propriété [float](#), les blocs se positionnent les uns à coté des autres.

L'idée ici va donc être de créer un bloc 'conteneur', qu'on positionnera à gauche du contenu (à l'aide de la propriété [float](#) donc). Et dans ce bloc, nous allons en placer deux autres, 'menuhaut' et 'menubas', qui eux vont se positionner l'un au dessus de l'autre.

Vous me suivez toujours ? Très bien, alors passons à la pratique... Nous allons utiliser le bloc 'menu' comme conteneur. Inutile donc de changer ses propriétés puisqu'il est déjà bien placé.

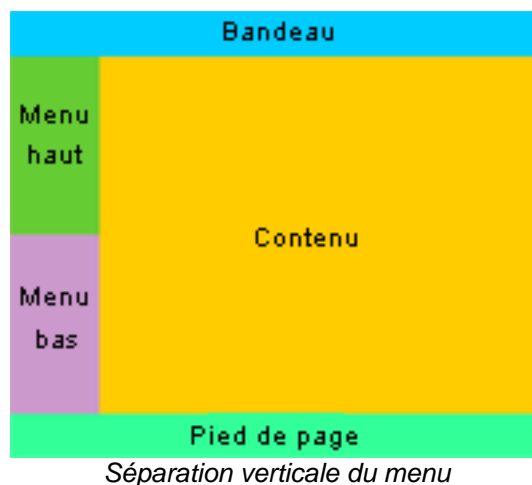
Propriétés des deux nouveaux blocs dans notre CSS :

```
div#menuhaut {  
    width:100px;  
    height:200px;  
    background-color:#66CC33;  
}  
div#menubas {  
    width:100px;  
    height:200px;  
    background-color:#CC99CC;  
}
```

Pas de propriétés [float](#) ici donc, puisque nous voulons que les deux blocs se superposent. Il ne reste plus qu'à demander à notre page de nous afficher ces deux nouveaux blocs, à l'endroit désiré :

```
<div id="menu">  
    <div id="menuhaut">Menu haut</div>  
    <div id="menubas">Menu bas</div>  
</div>
```

Et là, roulements de tambour... Merveilleux, exactement le résultat escompté :





[Télécharger les codes-source de l'exemple](#)



#### IV - Et un encart de news en haut à droite de mon contenu, sans que ça prenne toute la hauteur, c'est possible ?

Et oui !!! Décidément, le CSS, c'est surpuissant. Nous allons donc définir un nouveau bloc, appelé 'blocnews' (si c'est pas original ça ...).

Cette fois encore, nous allons faire appel à la propriété **float**.

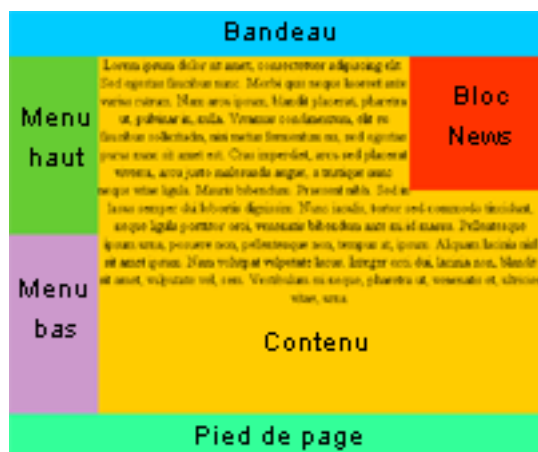
```
div#blocnews {
    float:right;
    width:150px;
    height:150px;
    background-color:#FF3300;
}
```

Rien de nouveau sous le soleil dans la définition des propriétés de notre élément, tout est connu. Passons maintenant à son intégration à notre page :

```
<div id="contenu">
  <div id="blocnews">Bloc News</div>
  Ceci est le contenu
</div>
```

La seule chose à laquelle il faut faire attention, c'est l'imbrication des différents éléments, qui, si elle est mal faite, peut vous pourrir la mise en page.

Si vous remplacez le texte 'Ceci est le contenu' dans le bloc contenu par un texte très long, vous vous apercevrez que le texte se décale pour suivre le contour du bloc news...



*Et hop ! un bloc de news ajouté*



[Télécharger les codes-source de l'exemple](#)