



doing it with
style

LES FEUILLES DE STYLES

C.S.S

Partie 3 sur 3

Le principe des « boîtes » et notions de « flux »

Pour représenter le positionnement en flux normal, on peut imaginer le navigateur parcourant (logiquement) la page de code HTML du début à la fin et retranscrivant son contenu au fur et à mesure des balises rencontrées.

http://openweb.eu.org/articles/initiation_flux/

Boîte de type bloc en flux normal :

Par défaut, les boîtes de type bloc seront affichées dans une succession verticale. Prenons par exemple deux blocs différenciés par leur couleur :

En *HTML* :

```
<p class="jaune">Une boîte jaune</p> <p class="verte">Une boîte verte</p>
```

En *CSS* :

```
.jaune { background-color: #ffff00; } .verte { background-color: #00ff00; }
```

Une boîte jaune

Une boîte verte

Les principaux éléments créant des boîtes bloc sont :

- * l'élément `div`
- * les titres `h1`, `h2`, `h3`, `h4`, `h5`, `h6`
- * le paragraphe `p`
- * Les listes et éléments de liste `ul`, `ol`

Boîte de type « en ligne » :

Par défaut, les boîtes de type en-ligne sont affichées dans une succession horizontale. Prenons par exemple deux portions de texte différenciées par leur couleur :

En *HTML* :

```
<p> <span class="jaune">Une boîte jaune</span> <span class="verte">Une boîte verte</span> </p>
```

En *CSS* :

```
.jaune { background-color: #ffff00; } .verte { background-color: #00ff00; }
```

Une boîte jauneUne boîte verte

Les principaux éléments créant des boîtes en ligne sont :

- * l'élément `span` ;
- * le lien `a` ;
- * L'image `img` ;
- * `strong` ;

Tout élément peut être "reclassé" dans la famille opposée grâce à la propriété **display**.

Il est préférable, *dans la mesure du possible*, de laisser à un élément son comportement « naturel ».

Position des éléments

Chaque bloc peut être placé de différentes façons à l'intérieur de la page web :

- par rapport à d'autres blocs
- à un endroit précis *du bloc qui le contient*
- à un emplacement fixe sur la page.

La position relative:

Positionnement par rapport à la « position qu'elle aurait »

La position absolue:

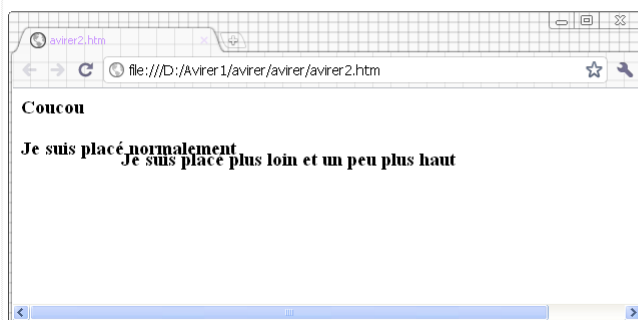
Position « en dur » par rapport à la boîte conteneur

*A connaître :
propriété float*

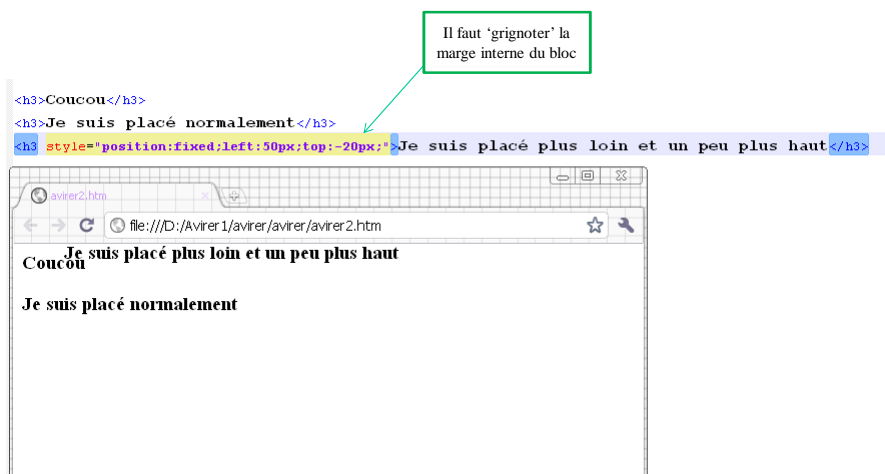
La propriété float est utilisée pour « habiller » des éléments (exemple :
enrouler" un texte autour d'un élément graphique)

La position relative:

```
<h3>Coucou</h3>
<h3>Je suis placé normalement</h3>
<h3 style="position:relative;left:100px;top:-30px;">Je suis placé plus loin et un peu plus haut</h3>
```



La position absolue:



Remarques pour la position absolue

Un bloc positionné se place par rapport au bloc qui le contient (son conteneur), mais seulement si celui-ci est lui-même positionné, que ce soit en position relative ou absolue. Si ce n'est pas le cas (conteneur dans le flux ou flottant), le bloc à positionner remonte de parent en parent jusqu'au premier bloc positionné (jusqu'à <body> s'il n'y en a pas) et se place par rapport à lui. Pour placer en position absolue un bloc dans un autre, il est donc important de vérifier si le conteneur est lui-même positionné. Si ce n'est pas le cas, il faut donner à ce bloc conteneur une position relative avec un décalage nul, ce qui ne modifiera pas sa position.

Il y a 2 façons de positionner en absolue :
absolute et fixed

absolute L'élément est positionné par rapport à son parent

fixed L'élément est positionné par rapport au navigateur et va donc rester fixe même en cas de déplacement avec les ascenseurs.

Démonstration

http://www.w3schools.com/cssref/playit.asp?filename=playcss_position&preval=fixed

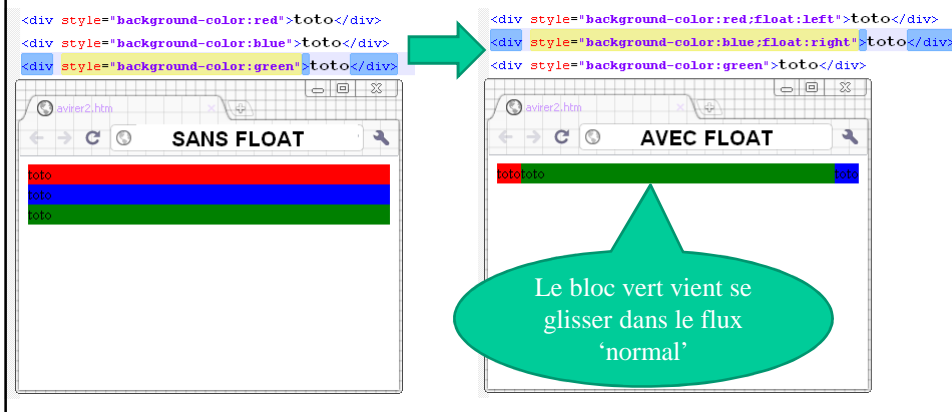
Enregistré en page locale *absolute-fixed-demo.htm*

La propriété float

Un élément peut être déclaré « flottant » à gauche ou à droite, avec la propriété float qui s'écrit alors respectivement float: left; ou float: right;

Le bloc est placé le plus à gauche ou le plus à droite possible.

Le contenu qui suit encadre alors cette boîte flottante,



La propriété float

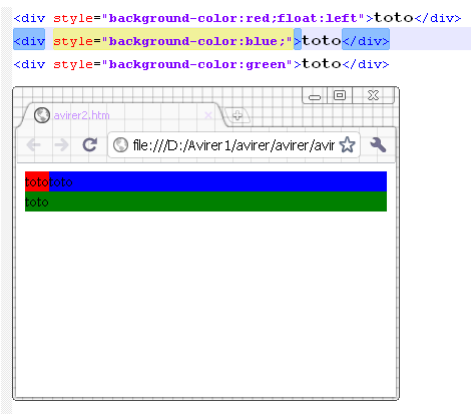
Exemple :

Le bloc rouge est en float:left

Le bloc bleu prend sa place dans le flux normal mais est « gêné » par le bloc rouge.

Il se place donc juste à côté et prend toute la largeur par défaut

Le bloc vert prend sa place dans le flux

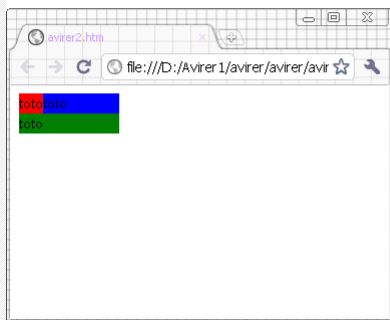


La propriété float

Exemple :

Idem précédent mais on donne une largeur au bloc bleu et une largeur au bloc vert

```
<div style="background-color:red;float:left">toto</div>
<div style="background-color:blue;width:100px">toto</div>
<div style="background-color:green;width:100px">toto</div>
```



Bien qu'ayant la même largeur, le bloc vert est plus large.

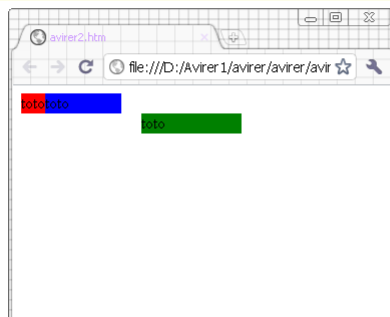
On en déduit que la largeur est la largeur par rapport au flux 'normal', comme si le bloc rouge n'existait pas

La propriété float

Exemple :

Idem précédent mais on donne décale le bloc vert sur la droite de 120 px

```
<div style="background-color:red;float:left">toto</div>
<div style="background-color:blue;width:100px">toto</div>
<div style="background-color:green;width:100px;position:relative;left:120px">toto</div>
```



« L'une des propriétés les plus utilisées en matière de CSS, float, avec ses valeurs left, right et none, reste celle dont le comportement est le moins prédictible par les designers Web même non débutants.

Combiné à la propriété clear, et ses valeurs left, right, both et none, cela donne parfois des arrangements de règles *hasardeuses* dont l'affichage final est le résultat avant tout d'une suite de **tâtonnements et de rechargements de page**. »

<http://www.journaldunet.com/developpeur/tutoriel/css/061130-css-comprendre-float-clear.shtml>

Exemple


```
<style>
#global
{
    margin-left: auto; margin-right: auto;
    width: 80%; /* largeur obligatoire pour être centré */
    background-color: lime
}
.float {
    width: 40%; float: left;
    margin: 10px;
    background-color: orange;
}
</style>
</head>
<body>
<div id="global">
<div class="float">
11111111<br>11111111
</div>
<div class="float">
22222222<br>22222222<br>
</div>
<div class="float">
33333333<br>33333333
</div>
<div class="float">
44444444<br>44444444
</div>
</div>

</body>
</html>
```

Les boîtes vont se succéder horizontalement sauf quand elles vont rencontrer le bord droit du conteneur parent

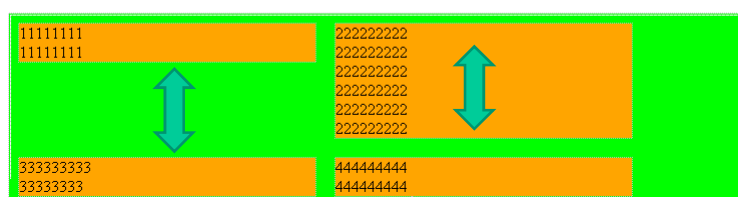
11111111	22222222	
11111111	22222222	
33333333	44444444	
33333333	44444444	


```
<div id="global">  
  <div class="float">  
    11111111<br>11111111  
  </div>  
  <div class="float">  
    2222222222<br>2222222222<br>2222222222<br>2222222222<br>2222222222<br>2222222222<br>2222222222<br>2222222222<br>  
  </div>  
  <div class="float">  
    3333333333<br>3333333333  
  </div>  
  <div class="float">  
    4444444444<br>4444444444  
  </div>  
</div>
```



```
</body>  
</html>
```

Problème ?!



Solution possible : Séparer le contenu en 2 colonnes qui vont contenir les blocs

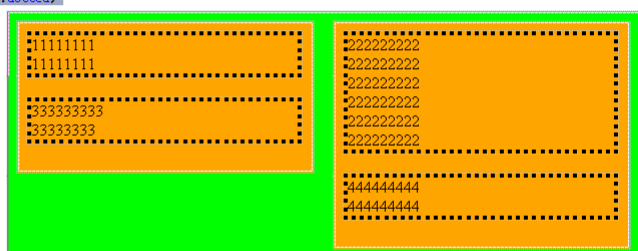
```

.float {
  width: 40%; float: left;
  margin: 10px;
  background:orange;
}

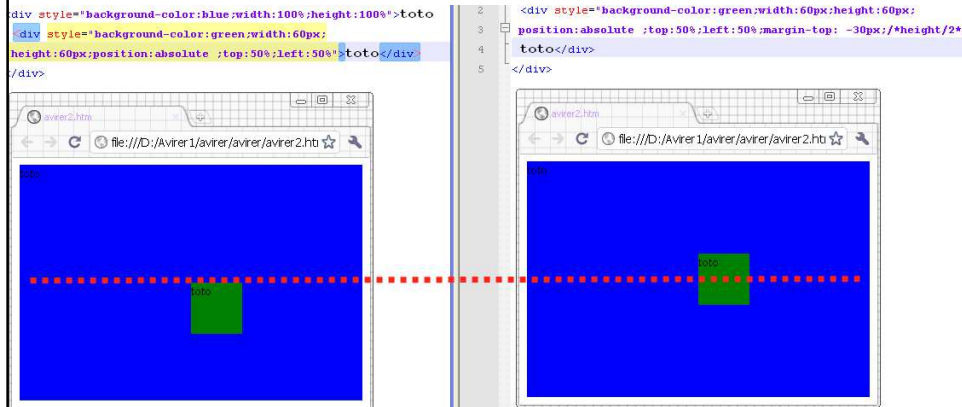
.bloccfinal {
  width: 90%; float: left;
  margin: 10px; border-style:dotted;
  background:orange;
}
</style>
</head>
<body>
<div id="global">
  <div class="float">
    <div class="bloccfinal">
      llllllll<b>llllllll
    </div>
    <div class="bloccfinal">
      33333333<b>33333333
    </div>
  </div>

  <div class="float">
    <div class="bloccfinal">
      22222222<b>22222222<b>22222222
    </div>
    <div class="bloccfinal">
      44444444<b>44444444
    </div>
  </div>
</div>
</div>

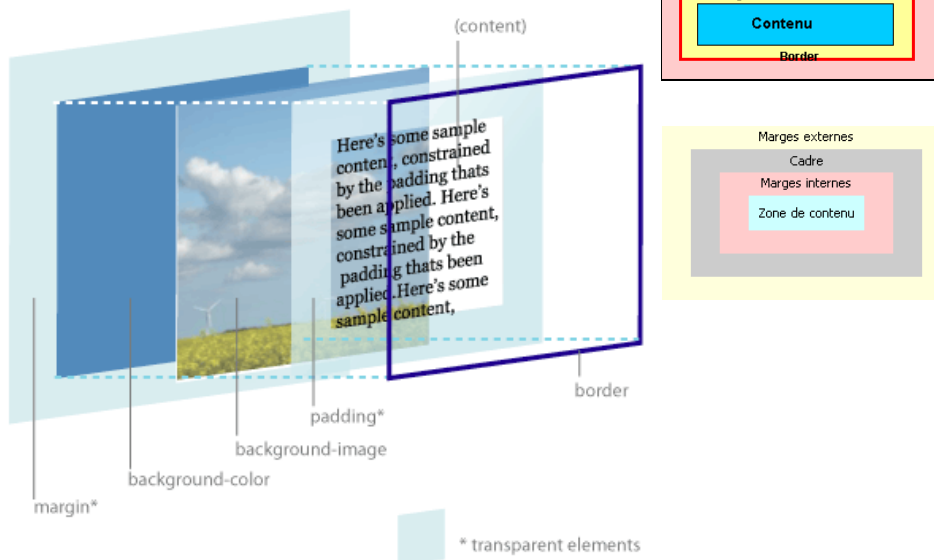
```



Comment centrer verticalement un bloc ?



Marges et boîtes

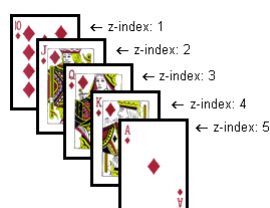
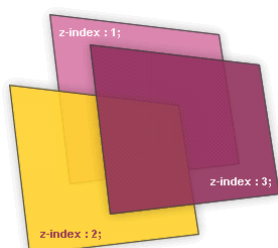


Pas simple ... d'autant que IE ne respecte pas certains principes de marge

La propriété z-index

Niveau d'empilement des blocs :

Lorsque plusieurs blocs sont superposés, ils s'empilent suivant l'ordre de leur arrivée dans le code HTML.
Cet arrangement peut toutefois être modifié, grâce à la propriété z-index



T.P 1/2

LIRE LES ARTICLES :

« Initiation au positionnement CSS : 1.flux et position relative »

« Initiation au positionnement CSS : 2.position float »

« Initiation au positionnement CSS : 3. position absolue et fixe »

A voir !

<http://www.buvetteetudiants.com/cours/administrator/html-css/positionn>

<http://www.barelyfitz.com/screencast/html-training/css/positioning/>