

**Savez-vous ce qui importe le plus aux yeux des consommateurs français ?**

Nous avons consulté 501 cyberconsommateurs afin de découvrir leurs attentes. Faites le premier pas pour leur donner ce qu'ils demandent.

[En savoir plus](#)

PUBLI-INFO

**CSS : comprendre le fonctionnement de float et clear**

*Propriété reine du placement des éléments d'un site, float a des conséquences encore imprévisibles pour qui n'a pas intégré sa philosophie. clear leur sera d'une aide précieuse. Explications et démonstrations en images.*

**30/11/2006** Pourtant 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 à d'une suite de tâtonnements et de rechargements de page.



Forum

[Réagissez](#) dans les forums de JDN Développeurs

Comprendre le comportement de `float` implique une compréhension plus globale de la philosophie des CSS, notamment celle du flux. Les CSS sont en effet conçues pour afficher les éléments au sein d'un flux : pour simplifier, les éléments en ligne, ou *inline*, sont affichés les uns à la suite des autres, tandis que les éléments conteneurs, ou *block*, amènent un retour à la ligne après leur affichage. Le texte et les balises de présentations (`b`, `i`, `span`, ...) sont des éléments inline, les images et balises de regroupement (`p`, `div`, `form`, ...) sont des éléments block. Le flux fait en sorte que ces éléments s'affichent naturellement, les uns à la suite des autres, en respectant les spécificités de chaque type.



Du code XHTML avec les bordures stylées

`float` permet de sortir un élément du flux, en cela que l'on peut préciser si l'élément ciblé, par exemple une image ou une colonne de contenu, s'affiche à gauche ou à droite. Et c'est ici trop souvent que le bât blesse : les designers voient souvent uniquement cette caractéristique de `float`, à savoir l'affichage de l'élément à gauche ou à droite. Ce faisant, ils oublient une autre caractéristique, implicite mais peut-être plus importante encore : le reste du flux, lui, sera décalé à droite de l'élément flottant, et poursuivra en dessous selon sa

longueur.

De fait, si l'on peut facilement déduire le comportement de l'élément flottant, celui du reste du flux devra être deviné par le designer, qui devra prévoir l'écoulement des éléments inline et block bien à l'avance. C'est pourquoi il est nécessaire de préciser la taille d'un élément flottant, ainsi que de prévoir la place prise par le reste du flux.

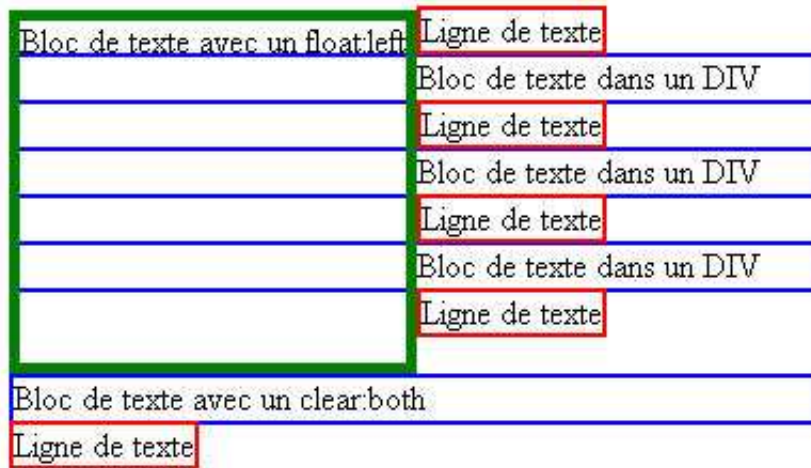


Le même code, avec la propriété float:left appliquée au tout premier bloc de texte



Une ligne de texte a maintenant le propriété float:right.

Le comportement inattendu du reste du flux se retrouve souvent sous la forme d'autres éléments flottants qui n'occupent pas la place voulue. C'est ici que la propriété `clear` entre en jeu. Celle-ci contrôle le comportement des éléments qui suivent, dans le flux, un élément flottant. Par défaut, l'élément standard suivant un élément flottant s'affichera dans l'espace laissé par la largeur du flottant. Ce comportement n'étant pas forcément voulu, `clear` a été conçue pour préciser que cet élément ne sera pas influencé par le comportement d'un flottant. Cette précision peut se faire sur chaque côté ou sur les deux à la fois. `clear` permet de libérer l'élément de l'influence du flottant. Tout le contenu situé entre le flottant et le libéré restera sous l'influence du flottant.



Même code, cette fois avec la propriété clear:both appliquée au dernier bloc de texte

Le code de la dernière image, ramené aux fondamentaux :

```

1. <style type="text/css">
2. span
3. div
4. .left
5. .right
6. .clear
7. </style>
8.
9. <div class="left">Bloc de texte avec un float:left</div>
10. <span>Ligne de texte</span>
11. <div>Bloc de texte dans un DIV</div>
12. <span>Ligne de texte</span>
13. <div>Bloc de texte dans un DIV</div>
14. <span>Ligne de texte</span>
15. <div>Bloc de texte dans un DIV</div>
16. <span>Ligne de texte</span>
17. <div class="clear">Bloc de texte avec un clear:both</div>
18. <span>Ligne de texte</span>

```

[Xavier Borderie, JDN Développeurs ...](#)

Copyright 2006 Benchmark Group - 69-71 avenue Pierre Grenier, 92517 Boulogne Billancourt Cedex, FRANCE

[Lancer l'impression](#)