



# Franck CELLIER

Formateur en Informatique

Technicien Qualité d'Organisme de Formation

<http://cellierfranck.alwaysdata.net>



## MS LANGAGE & DEVELOPPEMENT POUR LE WEB

### LES PROPRIÉTÉS « POSITION » ET « FLOAT »

#### A. NOTION DE FLUX

Le flux d'un document pourrait se définir comme étant le comportement naturel d'affichage des éléments d'une page web. Autrement dit, les éléments se succèdent dans l'ordre où ils sont déclarés dans le code HTML. Grossièrement, les éléments de séquençage, tels que h1, p ou ul, s'affichent logiquement par défaut les uns en-dessous des autres tandis que les éléments délimitant pour la plupart des portions de textes dans un but de gain sémantique s'affichent côte à côte. Nos deux groupes d'éléments peuvent être classés en deux familles:

- Les éléments de type block (h1, p, ul, ol, dl, table, blockquote, etc.)
- Les éléments de type inline (a, img, strong, abbr, etc.)

Un élément de type block peut s'identifier à une brique, à un gros "bloc" d'informations qu'on va pouvoir manipuler aisément. Il se différencie des éléments de type en ligne sur différents points, dont ceux-ci :

- Il occupe l'entièreté de la largeur de son conteneur
- Il permet l'attribution de marges verticales
- Il permet la modification de sa hauteur et largeur

Le premier point est **primordial**, nous utiliserons abondamment ce comportement par défaut bien pratique dans la création de nos mises en page. Notons également que tout élément peut être "reclassé" dans la famille opposée grâce à la propriété display.

Il faut garder à l'esprit qu'un élément dans le flux sera toujours plus simple à manipuler et plus flexible dans sa relation aux autres éléments. Gardez en tête que laisser à un élément son comportement naturel est *toujours* à privilégier. Autrement dit, on ne sortira un élément du flux que lorsqu'on ne pourra pas faire autrement. Bien entendu, le flux montre rapidement ses limites lorsqu'on envisage une mise en page un peu sophistiquée...

## B. LA PROPRIÉTÉ « POSITION »

La propriété `position` porte merveilleusement bien son nom puisqu'elle va nous permettre de gérer les... positions. C'est la propriété vers laquelle l'on se doit de se tourner en premier lieu dès que l'on considère une mise en page pour laquelle les possibilités du flux ne suffisent plus, et c'est là toute la raison d'être de cette puissante propriété.

### La position relative

La position relative (`position:relative`) permet de décaler un élément par rapport à une position de référence: celle qu'il avait dans le flux. Les éléments qui le suivent et le précèdent ne sont pas influencés par ce décalage puisqu'ils considèrent que l'élément est toujours dans le flux à sa position initiale. En pratique, ce comportement est rarement recherché bien qu'il puisse s'avérer utile dans certains cas.

Attribuer à un élément une position relative peut par contre être pratique, voire indispensable, dans d'autres situations dont les plus courantes sont sans nul doute les suivantes:

- Servir de référent à un élément enfant positionné en absolu (rappelons qu'un élément positionné absolument grâce aux propriétés `top`, `left`, ... le fera par rapport à la fenêtre du navigateur à défaut d'avoir un parent lui-même positionné)
- Bénéficier de la possibilité d'utiliser la propriété `z-index` pour gérer des superpositions d'éléments (propriété inopérante pour des éléments du flux)

### La position absolue

La position absolue (`position:absolute`) permet de ne pas dépendre de l'ordre dans lequel les éléments HTML sont déclarés dans le code, contrairement aux flottants que nous verrons plus tard.

La position absolue s'affranchit définitivement du cordon liant jusqu'alors intimement contenu et présentation. L'élément étant totalement extrait du flux, il ne dépend plus du tout des éléments qui le côtoient. Il faut voir le positionnement absolu comme étant une méthode radicale (mais puissante) qui retire tout à fait un élément du flux: il n'existe pour ainsi dire plus aux yeux des éléments qui, eux, restent dans le flux.

**Rappelons un point important concernant ce mode de positionnement : un élément positionné en absolu se réfère non pas à son parent direct, mais au premier ancêtre *positionné* qu'il rencontre.**

L'élément, n'étant plus dans le flux naturel, perd une de ses caractéristiques majeures qui est celle de recouvrir la totalité de la largeur disponible de l'élément parent.

Il est capital de noter qu'un élément bénéficiant d'une position absolue ne bougera pas de sa position initiale tant que l'une des propriétés `top`, `bottom`, `left` ou `right` n'a pas été précisée; il s'agit d'ailleurs là d'un comportement applicable à toutes les positions.

## La position fixe

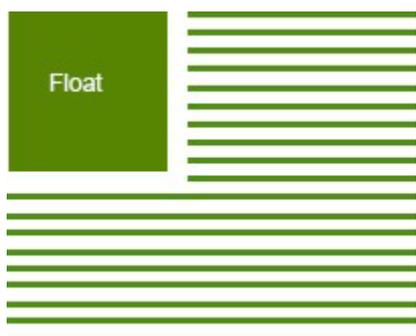
Le positionnement fixe (`position:fixed`) s'apparente au positionnement absolu, à l'exception des points suivants:

- Lorsque le positionnement est précisé (`top`, `right`, ...), l'élément est *toujours* positionné par rapport à la fenêtre du navigateur
- L'élément est fixé à un endroit et ne pourra se mouvoir, même lors de la présence d'une barre de défilement. En d'autres termes, la position initiale est fixée au chargement de la page, le fait qu'une éventuelle scrollbar puisse être utilisée n'a aucune influence sur le positionnement de l'élément: il ne bouge plus de la position initialement définie.

## La position statique

La position statique (`position:static`) correspond simplement à la valeur par défaut d'un élément du flux. Il n'y a que peu d'intérêt à la préciser, si ce n'est dans le but de rétablir dans le flux un élément en particulier parmi une série qui serait positionnée hors du flux.

## C. LA PROPRIÉTÉ « FLOAT »



La propriété `float` demeure sans doute, malgré des qualités intrinsèques évidentes, la propriété la plus mal comprise mais surtout, la plus mal employée des CSS. Sa simplicité d'utilisation apparente recèle pourtant certaines subtilités fondamentales et lui confère malheureusement un choix inconsciemment systématique pour toute disposition d'éléments en plusieurs colonnes.

### Rôle & détournement des flottants

La propriété `float` existe avant tout pour répondre à un besoin typographique précis: la création d'habillages. *Un habillage* est une pratique courante dans le média print consistant à "enrouler" un texte autour d'un élément (graphique ou texte); il était normal de retrouver cette ouverture de mise en page dans notre média favori.

À l'instar du positionnement absolu, un élément flottant adopte par défaut la largeur qu'occupe son contenu. Le principe de base est simple: un élément flottant est ôté partiellement du flux et placé à l'extrême gauche (`float:left`) ou droite (`float:right`) de son conteneur, forçant par la même occasion tout contenu du flux qui suit à l'envelopper. Deux objets flottants dans la même direction se rangeront côte à côte, seul un contenu demeuré dans le flux qui les succède immédiatement initiera l'habillage. C'est justement ce caractère hâtif de disposition adjacente de blocs qui est la cause d'une mauvaise et sur-utilisation de cette propriété trop souvent écartée de son application originale.

## Nettoyer les flottants

La propriété `clear` s'utilise conjointement aux `float` et permet à un élément (qui peut être d'ailleurs lui-même flottant) de ne plus subir le comportement d'habillage dicté par un objet flottant qui le précède directement et, par conséquent, de se caler en-dessous de ce dernier. Le `clear` autorise un nettoyage des flottants exclusivement à gauche (`clear:left`), à droite (`clear:right`) ou les deux simultanément (`clear:both`).