

Empêcher les flottants de dépasser de leur conteneur

Par défaut dans les navigateurs, les éléments positionnés grâce à la propriété CSS `float` (aussi appelés flottants, ou en anglais *floats*) sortent du flux de leur conteneur. Autrement dit, les flottants «dépassent». C'est ainsi qu'on se retrouve avec des conteneurs qui font zéro pixel de hauteur, des colonnes qui passent à travers un pied de page, et autres résultats peu souhaitables.

Par exemple, avec ce code HTML et CSS les «colonnes» flottantes dépasseront de leur conteneur:

```
<div id="conteneur">
  <div class="colonne">AAA</div>
  <div class="colonne">BBB</div>
  <div class="colonne">CCC</div>
</div>
```

```
#conteneur {
  height: auto; /* Par défaut. Prend la hauteur du contenu. */
  border: 2px solid red;
}
#conteneur > .colonne {
  width: 33%;
  float: left;
}
```

AAA BBB CCC

Comment bloquer ponctuellement le dépassement des flottants? Il existe plusieurs méthodes. Voici les trois principales, avec pour chacune une mention des principaux avantages et inconvénients, pour obtenir le résultat attendu :

AAA BBB CCC

La propriété `overflow`

La propriété CSS `overflow` définit comment un bloc doit se comporter lorsqu'un contenu dépasse de ce bloc. Les valeurs possibles sont `visible` (le contenu dépasse et est visible), `hidden` (le contenu qui dépasse est masqué), `scroll` (on affiche systématiquement des barres de défilement pour le bloc) et enfin `auto` (on affiche des barres de défilement *si un contenu dépasse*).

Cette propriété a aussi pour caractéristique, lorsqu'on utilise les valeurs autres que `visible`, de créer ce qu'on appelle un [contexte de formatage de bloc](http://www.w3.org/TR/CSS21/visuren.html#block-formatting) » <http://www.w3.org/TR/CSS21/visuren.html#block-formatting> . Et une des propriétés des contextes de

formatage de bloc, c'est d'empêcher le dépassement des flottants. C'est magique!

Voici comment on peut utiliser `overflow` pour éviter que les éléments flottants:

```
#conteneur {
  overflow: hidden; /* Empêche le dépassement des flottants. */
}
```

Avantage: technique efficace, tient en une ligne.

Inconvénient: lorsqu'on veut faire dépasser certains éléments du conteneur (par exemple un contenu court ou un élément de décoration), le `overflow:hidden` est gênant car il masque tout ce qui dépasse.

La propriété `clear`

La propriété CSS `clear` permet d'indiquer qu'un élément doit venir se placer en dessous des éléments flottants qui le précèdent. Les valeurs possibles sont `none` (par défaut), `left` (l'élément passe en dessous des éléments flottant à gauche), `right` (l'élément passe en dessous des éléments flottant à droite), et `both` (il passe en dessous de tout les flottants).

Pour bloquer le dépassement des flottants, on peut placer un élément vide juste avant la balise fermante du conteneur, et on lui applique un `clear:both`; (Attention, `clear` n'affecte que les éléments en `display:block`; , donc on utilisera de préférence un élément `div`, ou bien on rajoutera un `display:block`; dans les styles CSS.) En reprenant notre exemple, nous devons modifier le code HTML ainsi:

```
<div id="conteneur">
  <div class="colonne">...</div>
  <div class="colonne">...</div>
  <div class="colonne">...</div>
  <div class="clear"></div>
</div><--#conteneur-->
```

Notre code CSS devient:

```
#conteneur {
  /* Rien de spécial ici. */
}
#conteneur > .clear {
  clear: both;
  height: 0; overflow: hidden; /* Précaution pour IE 7 */
}
```

Avantage: technique éprouvée, pas ou peu d'effets indésirables.

Inconvénient: il faut ajouter un élément «inutile» dans le code HTML.

Utiliser `clear` sans modifier le code HTML (clearfix)

Pour finir, voici une variante intéressante de la technique du `clear: both`; . On peut éviter de rajouter un élément vide dans le code HTML si on simule cet élément directement en CSS. C'est possible en CSS 2.1 grâce à ce que CSS appelle le «contenu généré». On utilisera pour ça le pseudo-élément `:after` (notons qu'en CSS 3 on écrira `::after`, mais cette notation n'est pas comprise de tous les navigateurs actuels).

On repart donc de notre code HTML propre. Rien dans les poches, rien dans les manches:

```
<div id="conteneur">
  <div class="colonne">...</div>
  <div class="colonne">...</div>
  <div class="colonne">...</div>
</div><!--#conteneur-->
```

Et en CSS, on va générer un pseudo-élément qui va se rajouter à la fin de notre conteneur:

```
#conteneur {
  /* Rien de spécial ici. */
}
#conteneur:after {
  content: ""; /* Important, sinon l'élément n'est pas généré. */
  display: table;
  clear: both;
}
```

Avantage: pas d'élément rajouté dans le HTML, et pas ou peu d'effets indésirables.

Inconvénient: en inspectant une page avec les outils pour développeur (Firebug, Web Inspector, etc.), les pseudo-éléments ne sont pas visibles. Sans relire le fichier CSS lui-même, on ne peut pas savoir que cette technique est utilisée. Enfin, cette technique est compatible avec tous les navigateurs récents (dont IE 8), mais pas avec IE 7.

Le cas IE 7... voire IE 6

Toutes les techniques proposées sont compatibles avec toutes les versions récentes des navigateurs (en comptant IE8 dans les versions récentes). Par contre, certaines techniques ne sont pas compatibles avec IE 7 ou IE 6. Je récapitule, par compatibilité décroissante:

- `div` vide en `clear:both`, compatible IE 7 et IE 6;
- `overflow:hidden`, compatible IE 7 mais *pas* IE 6;
- pseudo-élément `:after` en `clear:both`, à partir d'IE 8 uniquement.

Une première solution pour le problème IE 6-7 c'est de choisir une technique compatible. Une autre solution c'est d'utiliser, en complément de ces techniques, le *hasLayout* — un comportement spécifique aux anciennes versions d'Internet Explorer (avant IE 8), qui a aussi pour caractéristique de bloquer le dépassement des flottants.

En gardant le même code HTML, la technique à base de `overflow:hidden` devient:

```
#conteneur {
  overflow: hidden; /* Empêche le dépassement des flottants. */
  zoom: 1; /* Idem pour IE 6-7 (active l'état "hasLayout") */
}
```

Et la technique du pseudo-élément `:after` devient:

```
#conteneur {
  zoom: 1; /* "hasLayout", empêche le dépassement
           des flottants dans IE 6-7 */
}
#conteneur:after {
  content: "";
  display: table;
```

```
clear: both;  
}
```

L'état *hasLayout* peut aussi, dans certaines circonstances particulières, causer des bugs de rendu... ou au contraire en corriger certains. Pour toute personne qui travaille sur du code CSS destiné aux anciennes versions d'Internet Explorer (avant IE 8), il est indispensable de lire [HasLayout et bugs de rendu dans Internet Explorer 6-7](http://www.alsacreations.com/article/lire/76-haslayout-internet-explorer.html) » <http://www.alsacreations.com/article/lire/76-haslayout-internet-explorer.html> .

Ressources

- <http://www.w3.org/TR/CSS2/visuren.html#floats> » <http://www.w3.org/TR/CSS2/visuren.html#floats>
- http://openweb.eu.org/articles/initiation_float/ » http://openweb.eu.org/articles/initiation_float/