Techniques et astuces pratiques pour une mise en page CSS

Par *Mark Newhouse*

Article original : *Practical CSS Layout Tips, Tricks and Techniques*

Auteur : *Mark Newhouse*

Date de parution : 2001

Traduction : Denis Poisson

Date de traduction : septembre 2002

Thèmes : *CSS*, *Web design*

Niveau : *Intermédiaire*

La campagne de mise à jour de navigateurs (*Browser Upgrade Initiative [http://www.webstandards.org/upgrade/]*) du *WaSP (Web Standards Project) [http://www.webstandards.org/]*), a encouragé de nombreux designers à s'aligner sur les Standards pour leurs pages Web, en utilisant les CSS *(Cascading Style Sheets)* au lieu de tableaux pour leur mise en page.

"Les tableaux sont morts..."

De nombreux designers web ont suivi *l'exemple [http://www.alistapart.com/stories/journey/]* de Jeffrey Zeldman en publiant des tutoriels pour nous aider à passer le cap du design sans tableaux. Leurs premiers efforts ont visé à créer une ou plusieurs colonnes en utilisant le positionnement CSS au lieu de tableaux, ce qui permet de séparer complètement la structure de la présentation. Ces techniques générales ont été documentées et archivées sur le site d'Eric Costello (*glish [http://www.glish.com/css/]*) ainsi que sur celui de Rob Chandanais (*Blue Robot [http://www.bluerobot.com /web/layouts/]*).

D'autres ont suivi leur exemple, comme les *Box lessons [http://www.thenoodleincident.com/tutorials/box_lesson/index.html]* d'Owen Briggs ou le *"hack"* de Tantek Çelik, *détaillé par Eric Costello [http://www.glish.com/css/hacks.asp]* et *expliqué par Tantek lui-même [http://www.tantek.com/CSS/Examples /boxmodelhack.html]*. Sur le Web, *Dotfile [http://dotfile.net /ala_list.php]* et Web Nouveau (un site qui n'existe plus - NDT) font la liste de centaines de sites conçus avec une mise en forme CSS.

"...vivent les tableaux !"

Mais bien que ces excellentes présentations nous permettent de créer une mise en page en utilisant uniquement le positionnement CSS, d'autres questions pratiques se posent, lorsqu'en tant que designers web, nous sommes confrontés à un problème trivial à résoudre avec des tableaux, mais moins évident avec CSS. Une question de cet ordre fut posée sur la mailing-list *Webdesign-L [http://www.webdesign-l.com/]*, dont le titre était *« Tables are dead... long live tables. »*

La question

Supposez que vous vouliez avoir une page avec des images vignettes qui soient liées aux versions normales de ces images - un genre de page plutôt courant sur le Web. Supposez maintenant que chaque image contienne une courte description que l'on désire voir apparaître centrée sous l'image. Enfin, afin d'utiliser au mieux l'espace

Mark Newhouse crée des sites depuis 1996, et met à profit ses compétences de professeur pour présenter les techniques digitales au plus grand nombre. Auteur et présentateur régulier, il a écrit de nombreux articles techniques pour différents sites web, et se pose comme la force créative derrière le populaire site de ressources CSS, *Real World Style*.

Articles du même auteur

Domptez vos puces, dressez des listes

Feuilles de Styles en Cascades, Promesse contre Réalité, et un regard vers le Futur

d'affichage de votre navigateur, vous désirez aligner ces paires d'images/titres en ligne sur l'écran, mais de manière à ce qu'elles se réalignent lorsque que la largeur de la fenêtre du navigateur change (design fluide). Cette dernière spécification nous emmène hors des sentiers battus des tableaux, vers le royaume des CSS...

Un pas après l'autre

Regardons cela pas-à-pas. La première étape consiste à centrer les titres sous les vignettes. Cela est relativement simple : dans votre page HTML vous placez votre image, suivie d'un retour à la ligne (`<br />`), puis vous placez votre titre dans un paragraphe centré via CSS.

La prochaine étape consistera à aligner ces paire d'images/titres dans la fenêtre du navigateur. Si l'on utilisait des tableaux pour la mise en page, chacune de ces paires serait rangée dans un `td`. Puisqu'on utilise CSS, nous les mettrons dans des balises `div`, que nous alignerons horizontalement dans la fenêtre en utilisant un `float` à gauche dans la feuille de styles.

Voilà à quoi ressemblera la feuille de styles à ce moment :

```
div.float {
      float: left;
 }

div.float p {
      text-align: center;
  }
```

Quant au HTML :

```
<div class="float">
 <img src="image1.gif" width="100" height="100"
 alt="image 1" /><br />
 <p>légende 1</p>
</div>

<div class="float">
 <img src="image2.gif" width="100" height="100"
 alt="image 2" /><br />
 <p>légende 2</p>
</div>

<div class="float">

 <img src="image3.gif" width="100" height="100"
 alt="image 3" /><br />
 <p>légende 3</p>
</div>
```

Et dans le navigateur :

Voici la liste des dix articles les plus récents en rapport avec cet article :

Intermédiaire

Comment vous assurer que vos emails HTML s'affichent correctement et arrivent à bon port ?

Javascript non-intrusif, chapitre 4 : l'appel des scripts de la forêt

Javascript non-intrusif, chapitre 2 : comment retrouver les éléments que l'on veut manipuler

Javascript non-intrusif, chapitre 1 : le grand nettoyage !

Définir des tailles de police dynamiques cohérentes avec CSS

Contraste et sens

Bien utiliser le texte alternatif

Les Super-Sélecteurs !

Des CSS plus efficaces grâce aux raccourcis

Les déclarations « Doctype » et les en-têtes « Content-type »

CSS

Comment vous assurer que vos emails HTML s'affichent correctement et arrivent à bon port ?

On revoit les bases : la propriété « background »

Définir des tailles de police dynamiques cohérentes avec CSS

Les Super-Sélecteurs !

Des CSS plus efficaces grâce aux raccourcis

Design de poche : Porter votre site web au petit écran

CSS : on reprend tout à zéro ! (15ème épisode)

CSS : on reprend tout à zéro ! (14ème épisode)

CSS : on reprend tout à zéro ! (13ème épisode)

CSS : on reprend tout à zéro !

légende 1

légende 2

légende 3

La prochaine étape ne peut être résolue qu'avec les CSS. Nous voulons que les paires d'images/titres se rangent à la ligne suivante si elles sont trop nombreuses pour tenir toutes dans la fenêtre du navigateur. En fait, ce problème a déjà été réglé par l'utilisation d'un `float` à gauche sur les `div`. Si nous dupliquons quelques-unes de ces vignettes, celles-ci se rangeront à la ligne (pour en faire l'expérience, changez la largeur de votre fenêtre de navigateur) :

légende 1

légende 2

légende 3

légende 1

légende 2

légende 3

Maintenant, supposons que vous ayez plus d'une catégorie de vignettes à afficher dans votre page, et que vous désiriez les regrouper visuellement, par un arrière-plan et/ou une bordure. Il suffit alors de les entourer d'une balise `div` :

```
div.container {
    background-color: #D7E7F2;
   border: 1px solid #287CB1;
 }
```

Malheureusement, cela pose un nouveau problème. Quand on utilise `float` sur un élément avec CSS, il n'occupe plus "d'espace", et du coup l'arrière-plan et la bordure apparaissent au-dessus des images au lieu de les entourer. On doit donc ajouter un contenu différent des `div` en `float` dans le `div` d'encadrement. Par exemple, un `div` d'espacement :

```
div.spacer {
      clear: both;
 }
```

Maintenant le HTML apparaît comme suit (remarquez qu'il y a des `div` d'espacement au début et à la fin du `div` d'encadrement) :

```
<div class="container">
<div class="spacer">
  
</div>

<div class="float">
 <img src="image1.gif" width="100" height="100"
 alt="image 1" /><br />
 <p>légende 1</p>

</div>

<div class="float">
 <img src="image2.gif" width="100" height="100"
 alt="image 2" /><br />
 <p>légende 2</p>
</div>

<div class="float">
 <img src="image3.gif" width="100" height="100"
 alt="image 3" /><br />
 <p>légende 3</p>
</div>

<div class="spacer">
  

</div>

</div>
```

Ce qui nous donne la présentation suivante:

légende 1 légende 2 légende 3

Basé sur le code de *sam marshall [http://www.leafdigital.com/]*.

`div` imbriqués, tableaux imbriqués, quelle différence ?

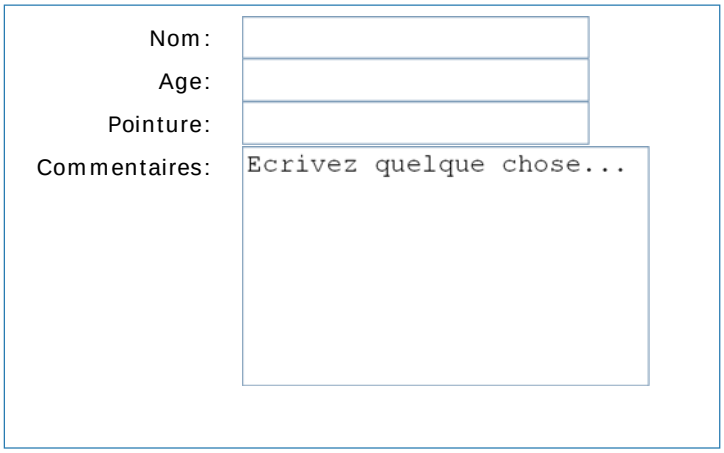
Nous avons maintenant tout un tas de `div` imbriqués. En quoi cela est-il meilleur que des imbrications de tableaux ? Cela vient de la façon dont il était originellement prévu d'utiliser ces balises. Les `div` impliquent un regroupement logique, ou structuré. Même imbriqués, ils représentent un balisage structuré. Dans notre exemple, nous avons regroupé les vignettes avec leurs titres (premier niveau de structure), puis regroupé ces paires avec d'autres paires similaires (second niveau). Ce genre de regroupement structurel convient parfaitement à l'emploi de balises `div`.

Par contre, les tableaux impliquent une relation entre les titres de colonne et/ou de ligne, et l'information contenue dans les cellules du tableau. En les utilisant pour la mise en page, on perd la sémantique structurelle d'un tableau. On retourne à une utilisation de HTML comme outil de présentation et non de structuration. Et imbriquer des tableaux dans d'autres ne fait qu'aggraver le problème.

`form` et fonction

Une autre utilisation fréquente des tableaux en présentation sert à aligner les éléments `form` avec leur vignette. Bien qu'on puisse défendre l'opinion qu'il s'agit là d'une utilisation appropriée des tableaux, les techniques CSS décrites ci-dessous pourront être utilisées pour d'autres besoins de présentation similaires, comme nous allons le voir.

Une présentation de formulaire typique met les vignettes à gauche en les alignant vers la droite, et les champs à droite en les alignant vers la gauche. Le tout étant centré à la jonction du milieu :



Basé sur un concept original d'*Eric Meyer [http://www.meyerweb.com/]*.

Le formulaire ci-dessus a été créé sans l'aide de tableaux. Encore une fois, nous utilisons `float` pour générer le positionnement. L'astuce consiste à créer un `div` qui fonctionne comme une ligne du tableau que nous avons l'habitude d'utiliser. Ensuite il nous suffit de créer deux `span` : un pour les vignettes et l'autre pour les champs du formulaire. On place l'un en `float` droit, et l'autre en `float` gauche, puis on aligne le texte du `span` de la vignette à droite, et celui du `span` du champ à gauche.

La feuille de style ressemble à cela :

```
div.row {
      clear: both;
       padding-top: 10px;
 }

div.row span.label {
      float: left;
      width: 100px;
      text-align: right;
 }

div.row span.formw {
       float: right;
      width: 335px;
      text-align: left;
 }
```

Les styles ci-dessus définissent également une largeur pour les `span`. Cette dernière peut être absolue comme dans l'exemple, ou en pourcentages dont la somme serait de 100% ou moins, selon le padding et les bordures (ainsi que la présentation générale utilisée). Dans notre exemple j'ai imbriqué le formulaire dans un autre `div` afin de lui donner une bordure et un arrière-plan.

Le HTML de l'exemple :

```
<div style="width: 350px; background-color: #D7E7F2; borde
  margin: 0px auto;">
 <form>
  <div class="row">

   <span class="label">Nom :</span>
   <span class="formw"><input type="text" size="25" /></sp
  </div>
  <div class="row">
   <span class="label">Age :</span>

   <span class="formw"><input type="text" size="25" /></sp
  </div>
  <div class="row">
   <span class="label">Pointure :</span>
   <span class="formw"><input type="text" size="25" /></sp

  </div>
  <div class="row">
   <span class="label">Commentaire :</span>
   <span class="formw">
    <textarea cols="25" rows="8">

    Ecrivez quelquechose...
    </textarea>
   </span>
  </div>
 <div class="spacer">
  
 </div>

 </form>

</div>
```

Devant et au centre

Vous avez sans doute remarqué que le style du `div` d'encadrement contenait `margin: 0px auto;`. Cette ligne doit centrer ce `div` de 400 pixels dans les navigateurs conformes aux standards. Certains navigateurs, comme IE5.x pour Windows, n'implémentent pas cette fonction, mais par contre centrent (incorrectement) les `div` dont l'attribut `text-align` prend la valeur `center`. Pour centrer un `div` dans de tels navigateurs, vous pouvez inclure un `div` avec `margin: auto` (et `text-align: left` pour aligner le texte correctement) dans un autre `div` avec `text-align` en `center`. Vous pouvez aussi aller voir le *Layout Reservoir* de Rob Chandanais pour plus de *détails [http://bluerobot.com/web/css/center1.html]*, ainsi que *d'autres techniques [http://bluerobot.com/web/css/center2.html]* de centrage.

De part et d'autre

Il existe une autre mise en page similaire (mais inverse) qui s'implémente généralement avec des tableaux. Il s'agit de placer des éléments de part et d'autre de la page, comme par exemple lorsque vous désirez placer un logo en haut à droite de votre page, et des éléments de navigation en haut à gauche :

Home > Products **[logo]**

Ici nous utiliserons le même `div.row` que précédemment, mais avec d'autres `span` que pour aligner les éléments de formulaire avec leurs vignettes. Le `span` de gauche sera en `float` à gauche, avec le texte aligné à gauche, et celui de droite en `float` à droite, avec le texte aligné à droite.

Le CSS :

```
div.row span.left {
      float: left;
      text-align: left;
      font-weight: bold;
      color: #000;
      width: 49%;
 }

div.row span.right {
      float: right;
      text-align: right;
      font-weight: bold;
      color: #000;
      width: 49%;
 }
```

Le HTML :

```
<div style="width: 90%; background-color: #666;
 border: 1px solid #333; padding: 0px; margin: 0px auto;">
 <div class="spacer"></div>
 <div class="row">
  <span class="left">Home > Products</span>

  <span class="right">[logo]</span>
 </div>
 <div class="spacer"></div>
</div>
```

Aide en CSS

`acronym` et `abbr` sont des balises utiles, bien que peu utilisées, qui se servent de l'attribut `title` pour développer l'acronyme ou l'abbréviation. La plupart des navigateurs ne font pourtant rien pour prévenir vos visiteurs qu'il existe quelque chose sous ces mots qui

pourrait les aider à en comprendre le sens. Entre alors en scène CSS.

En effet, vous pouvez dans votre feuille de style définir une bordure inférieure sur ces balises, qui vous permet d'attirer l'attention sur elles. Vous pouvez également utiliser CSS pour modifier l'apparence du curseur en curseur "d'aide" (généralement un point d'interrogation) sur les navigateurs qui implémentent cette spécification. Et vous n'êtes pas limités aux balises HTML. Vous pouvez créer une classe `.help` et utiliser des `span` pour donner plus d'informations sur les mots ou phrases qui pourraient laisser vos lecteurs perplexes.

Utilisez ce CSS :

```
abbr, acronym, .help {
 border-bottom: 1px dotted #333;
 cursor: help;
 }
```

en conjonction avec l'attribut `title` ou avec les balises `abbr` et `acronym`, afin de créer un effet de souligné différent du soulignement des hyperliens. En modifiant l'apparence du curseur, vous indiquez que ce mot n'est pas cliquable, et l'attribut `title` développe l'abbréviation ou l'acronyme. J'ai vu cela mis en oeuvre pour la première fois sur le site de *Sander Tekelenburg [http://www.euronet.nl/%7Etekelenb/]*.

Tous en rang !

J'ai appris comment mettre le `display` d'une liste en `inline` dans l'ouvrage *Cascading Style Sheets [http://www.awlonline.com /cseng/titles/0-201-41998-X/liebos/]* de Bos et Lie. Je l'ai vu mis en oeuvre pour la première fois sur le site de Christopher Schmitt *BabbleList [http://www.babblelist.com/]*. Cette technique prend une liste et l'affiche *inline*, c'est-à-dire horizontalement. Donc au lieu de :

- Premier élément
- Deuxième élément
- Troisième élément

on a :

Premier élément Deuxième élément Troisième élément

En ajoutant du padding et un effet de bordure on obtient :

Premier élément | Deuxième élément | Troisième élément

Le CSS :

```
li.inline {
      display: inline;
      padding-left: 3px;
      padding-right: 7px;
      border-right: 1px dotted #066;
 }

li.last {
      display: inline;
      padding-left: 3px;
      padding-right: 3px;
      border-right: 0px;
 }
```

Le HTML :

```
<ul>
 <li class="inline">Premier élément</li>
 <li class="inline">Deuxième élément</li>
 <li class="last">Troisième élément</li>

</ul>
```

La fin ? Ou le commencement ?

J'espère que le fait d'avoir partagé ces techniques avec vous vous encouragera à utiliser plus fréquemment une mise en page CSS dans vos publications Web, et que vous continuerez à découvrir et à partager avec d'autres de nouvelles techniques.
