

# Transformer une liste en barre de navigation

par Roger Johansson ([456 Berea Street](#)) Nicolas Vallée (traducteur) ([Home Page](#))

Date de publication : 04/01/2008

Cet article est la traduction de **Turning a list into a navigation bar** et a pour but de vous montrer comment créer, à partir d'une liste non numérotée, une barre de navigation.

- I - Genèse de l'article
- II - Le code HTML
- III - La feuille de style CSS
- IV - Remerciements

(1)

## I - Genèse de l'article

On m'a demandé quelques fois une description sur la méthode avec laquelle j'avais créé la barre de navigation qui est actuellement utilisée sur mon site. La feuille de style CSS utilisé n'était pas tellement évoluée, et je n'avais pas vraiment pensé à la décrire en détail, mais après m'être interrogé à ce sujet, j'ai décidé de commencer à rédiger.

J'ai nettoyé légèrement le *HTML* et le *CSS*, donc si vous comparez avec ce qui est maintenant utilisé sur le site, il y aura quelques petites différences. Dans l'hypothèse où j'aurais tout refait au moment où vous lirez cet article, jetez un oeil sur l'**exemple final** pour voir ce à quoi devait ressembler le menu au moment de la rédaction de cet article.

## II - Le code HTML

La structure du code HTML est très simple. Il s'agit d'une liste non ordonnée, où chaque lien est associé à un élément.

### Code (X)HTML

```
<ul id="nav">
  <li id="nav-home"><a href="#">Accueil</a></li>
  <li id="nav-about"><a href="#">A propos</a></li>
  <li id="nav-archive"><a href="#">Archives</a></li>
  <li id="nav-lab"><a href="#">Laboratoire</a></li>
  <li id="nav-reviews"><a href="#">Critiques</a></li>
  <li id="nav-contact"><a href="#">Contact</a></li>
</ul>
```

### Voir l'étape 1

Pourquoi utiliser une liste ? Parce que la barre de navigation, ou le menu, est une liste de liens. La meilleure (au sens sémantique) façon de représenter une liste de liens est d'utiliser un élément liste. Utiliser une liste possède aussi l'avantage de fournir une structure même si le CSS est désactivé.

A ce stade, avec aucune feuille de style CSS appliquée, la liste ressemble à n'importe quelle autre liste (normalement avec des points), dont l'aspect est défini par le comportement par défaut du navigateur.

J'ai donné des attributs **id** aux éléments de type *ul* et *li*. L'attribut **id** pour l'élément *ul* est utilisé par les règles CSS pour définir le style de la liste entière. Les éléments *li* ont des valeurs différentes pour **id** afin de permettre l'utilisation d'une règle CSS pour mettre en valeur le lien actuellement sélectionné. Cela se fera plus tard en spécifiant un **id** pour l'élément *body*.

## III - La feuille de style CSS

Je vais décrire étape par étape la feuille de style CSS que j'ai utilisée pour définir l'apparence de ma liste.

Tout d'abord, j'ai réduit à zéro les marges intérieures (padding) et extérieures (margin) de la liste, et j'ai demandé aux éléments de la liste de se disposer en ligne.

### Code CSS

```
#nav
{
  margin:0;
  padding:0;
}

#nav li
{
  display:inline;
  padding:0;
  margin:0;
}
```

### Voir l'étape 2

Ainsi tous les liens seront disposés les uns après les autres sur la même ligne, comme si la liste n'était pas là. Cela a également enlevé les points devant chaque élément de la liste, puisqu'ils ne sont présents que lorsque la propriété **display** vaut *list-item* (ce qui est la valeur par défaut pour les éléments d'une liste). On raconte que certains navigateurs afficheraient les points malgré la propriété *display:inline* appliquée aux éléments de la liste. Je n'ai jamais vu cela arrivé sur les navigateurs que j'ai testés, mais si vous souhaitez vous assurer qu'aucun navigateur n'affichera ces points, vous pouvez ajouter la règle *list-style-type:none* pour *#nav*.

Maintenant, il est temps de définir l'apparence des touches du menu. Je le fais en ajoutant un style aux liens, mais pas aux éléments de la liste. La raison de ce comportement est que je veux que l'intégralité de la zone de chaque touche soit cliquable. On commence par mettre un peu de couleur pour que les changements soient plus visibles :

### Code CSS

```
#nav a:link, #nav a:visited
{
  color:#000;
  background:#b2b580;
}
```

### Voir l'étape 3

Remarquez que je donne la même apparence aux liens déjà visités et aux autres liens.

L'étape suivante est d'augmenter le padding des liens :

### Code CSS

```
#nav a:link, #nav a:visited
```

**Code CSS**

```
{  
  color:#000;  
  background:#b2b580;  
  padding:20px 40px 4px 10px;  
}
```

 **Voir l'étape 4**

C'est un peu mieux. Mais il y a un problème potentiel qui n'est pas visible ici. Etant donné que les éléments sont disposés sur une ligne, leur padding vertical ne va pas s'ajouter à la hauteur de la ligne. Ce serait plus facile à observer si l'élément ul avait un arrière-plan, donc je vais lui ajouter une couleur et une image :

**Code CSS**

```
#nav  
{  
  margin:0;  
  padding:0;  
  background:#808259 url('nav_bg.jpg') 0 0 repeat-x;  
}
```

 **Voir l'étape 5**

Oups... Maintenant les liens débordent des éléments de la liste. Pour réparer cela, j'ai placé les liens dans des boîtes qui flottent vers la gauche. Pour cela, j'ai réglé à *auto* leur largeur (width), et je les ai fait se réduire pour s'ajuster à leur conteneur :

**Code CSS**

```
#nav a:link, #nav a:visited  
{  
  color:#000;  
  background:#b2b580;  
  padding:20px 40px 4px 10px;  
  float:left;  
  width:auto;  
}
```

 **Voir l'étape 6**

Ajouter la règle *display:block* aux règles CSS pour les liens les aurait également transformer en boîtes, mais étant donné qu'un élément flottant génère automatiquement une boîte, cela n'était pas nécessaire.

Comme vous avez pu le remarquer, l'arrière-plan a disparu lorsque les liens se sont mis à flotter. En effet, les éléments flottants sont sortis du flot du document, ce qui implique que la hauteur de l'élément ul les contenant a été réduite à zéro. Par conséquent, l'arrière-plan est là, mais il est invisible. Pour garder l'encapsulation des liens dans ul, je l'ai également fait flotter. Ainsi, j'ai défini sa largeur à 100%, ce qui le fait s'étendre sur l'intégralité de la fenêtre (sauf au niveau du padding que j'ai donné à l'élément body dans cet exemple) :

**Code CSS**

**Code CSS**

```
#nav
{
  margin:0;
  padding:0;
  background:#808259 url('nav_bg.jpg') 0 0 repeat-x;
  float:left;
  width:100%;
}
```

 **Voir l'étape 7**

Pour séparer visuellement les liens les uns des autres, je leur ai ajouté une bordure droite. Ensuite, pour donner au premier lien une bordure gauche identique, j'ai utilisé la pseudo-class *a:first-child* pour appliquer une règle uniquement au lien du tout premier élément la liste. J'ai aussi ajouté des bordures hautes et basses à l'élément ul :

**Code CSS**

```
#nav
{
  margin:0;
  padding:0;
  background:#808259 url('nav_bg.jpg') 0 0 repeat-x;
  float:left;
  width:100%;
  border:1px solid #42432d;
  border-width:1px 0;
}

#nav a:link, #nav a:visited
{
  color:#000;
  background:#b2b580;
  padding:20px 40px 4px 10px;
  float:left;
  width:auto;
  border-right:1px solid #42432d;
}

#nav li:first-child a
{
  border-left:1px solid #42432d;
}
```

La pseudo-classe *:first-child* n'est pas reconnue par le navigateur Internet Explorer pour Windows, donc le premier lien n'aura pas de bordure gauche sur ce navigateur. Dans ce cas, ce n'est pas un gros problème, donc j'ai laissé tel quel. Si cela est vraiment important pour vous, vous devrez ajouter une classe au premier élément de la liste (ou au lien qu'il contient), et l'utiliser pour donner une bordure gauche à ce lien.

 **Voir l'étape 8**

Ensuite, j'ai changé l'affichage dont le texte du lien en enlevant le soulignage, en le mettant en gras, en spécifiant une taille de caractère et une police de caractère différentes, en mettant en majuscules le texte, et en ajoutant une peu effet d'ombre. Cette ombre est créée par la propriété *text-shadow*, une propriété CSS3 qui n'est pour le moment supporté que par Safari et Omniweb :

## Code CSS

```
#nav a:link, #nav a:visited
{
  color:#000;
  background:#b2b580;
  padding:20px 40px 4px 10px;
  float:left;
  width:auto;
  border-right:1px solid #42432d;
  text-decoration:none;
  font:bold 1em/1em Arial, Helvetica, sans-serif;
  text-transform:uppercase;
  text-shadow: 2px 2px 2px #555;
}
```

 Voir l'étape 9

Pour donner un retour visuel quand je survole un lien, j'ai donné à leur état `:hover` des couleurs différentes pour le texte et l'arrière-plan :

## Code CSS

```
#nav a:hover
{
  color:#fff;
  background:#727454;
}
```

 Voir l'étape 10

Dans l'étape finale, j'ai ajouté des règles qui donneront un aspect différent au lien sélectionné, afin de montrer aux visiteurs où ils se situent dans le site.

Dans l'hypothèse où vous n'auriez jamais vu auparavant un exemple de spécification d'attribut `id` pour l'élément `body` afin de donner un aspect différent au lien de navigation courant, sachez que cela est effectué par les deux premières règles. Dans les exemples montrés dans cet article, j'ai donné à l'attribut `id` de l'élément `body` la valeur "home", ce qui fait du lien "Home" le lien courant. Si vous changez cet `id` en "about", le lien "About" sera le lien courant, et ainsi de suite...

J'ai également fait en sorte que le lien sélectionné reste le même lorsqu'on le survole avec la souris. On peut aussi se demander si l'élément courant dans le menu devrait carrément ne pas être un lien. Dans ce cas, j'ai choisi de le laisser sous la forme d'un lien et d'utiliser CSS pour enlever le retour visuel à son survol.

Pour donner aussi un retour visuel lorsqu'on clique sur l'un des liens, j'ai donné à l'état `:state` la même apparence que le lien sélectionné :

## Code CSS


```
#home #nav-home a, #about #nav-about a, #archive #nav-archive a, #lab #nav-lab a,
#reviews #nav-reviews a, #contact #nav-contact a
{
  background:#e35a00;
  color:#fff;
}
```

**Code CSS**

```
text-shadow:none;
}

#home #nav-home a:hover, #about #nav-about a:hover, #archive #nav-archive a:hover,
#lab #nav-lab a:hover, #reviews #nav-reviews a:hover, #contact #nav-contact a:hover
{
  background:#e35a00;
}

#nav a:active
{
  background:#e35a00;
  color:#fff;
}
```

 **Voir l'étape 11**, le menu de navigation final

C'est fini. Ce tutoriel pas-à-pas peut donner l'impression que l'ensemble est plus complexe qu'il ne l'est réellement. Regardez le source de l'exemple final pour observer l'intégralité des règles CSS. Bien entendu, à l'exception de quelques petits détails (la bordure gauche sur le premier lien, et les ombres du texte), cela fonctionne tel quel sur tous les navigateurs, même Internet Explorer (version 5 ou plus récent).

J'espère que vous avez pu suivre suffisamment bien pour être en mesure de créer votre propre menu de navigation. Les possibilités en ce qui concerne l'apparence sont presque infinies.

## IV - Remerciements

Tous mes remerciements à **RideKick** pour sa relecture.

1 :

Cet article a été publié à l'origine le 10 janvier 2005 et est encore visible à l'adresse suivante : **[Turning a list into a navigation bar](#)**