

# Créer une liste de texte survolable et cliquable

par Sam Rayner

Date de publication : 20/05/2008

Dernière mise à jour : 22/05/2008

Une technique émerge de plus en plus du net, c'est la création d'une liste, ayant un contenu plus que conséquent, survolable en utilisant des balises `a`. Ce tutoriel aura donc pour unique but de vous la présenter.

Introduction.....	3
L'approche.....	3
Quel est le problème ?.....	3
Un regard nouveau sur ces éléments.....	4
Astuces pour styliser votre Blocklist.....	5
Le Bon, le Méchant et IE.....	6
Remerciements.....	6

(1) (2)

## Introduction

Depuis peu, il existe une technique, que j'appellerai *blocklist*, est de plus en plus utilisée des concepteurs de sites web. Par *blocklist*, je veux dire une liste de liens, empilés verticalement, avec un contenu assez conséquent. Pour vous donner une idée, **Veerle Pieters** a gentiment acceptée, à titre d'exemple, de me laisser utiliser la rubrique **Approved** de son blog.

Je n'ai pas inclus tout le code nécessaire à l'obtention de la version finale, étant donné que c'est assez compliqué. Cependant, vous pouvez voir un **exemple** d'utilisation de cette technique et jeter un œil au code source. Même s'il est assez détaillé, je vous recommande de lire l'explication qui suit plutôt que de vous précipiter sur la démo; ça vous aidera à vous faire une idée sur le fonctionnement en cas de problèmes ou quand vous voudrez faire des modifications ultérieures.

## L'approche

Allez voir l'**implémentation** de Veerle : interagissez avec elle (survolez et cliquez sur les éléments de la liste). N'est-elle pas belle ? Cette large zone cliquable apporte du caractère à ce qui serait qu'un simple titre lié à une description positionnée à sa suite, sans oublier les avantages que cela apporte en termes d'accessibilité pour les personnes à motricité réduite.

Jetons un coup d'oeil au code :

```
<h2>Approved</h2>
<ul>
  <li><a href="#">
    FF Meta Serif<br>
    <em>A gorgeous, all-purpose typeface.</em><br>
    <span> December 11 at 09.47 am</span>
  </a></li>
</ul>
```

Actuellement, la seule manière de rendre le bloc entièrement cliquable est d'entourer chaque élément contenu dans le lien, c'est-à-dire le titre, la description et la date, d'une seule et unique balise **a**.

Comme **a** est un élément *inline*, il ne peut pas contenir de balises de type bloc (*block-level*), ce qui signifie que le titre ne peut être codé avec une entête (**hN**) et qu'il en va de même pour les paragraphes (**p**). Pour personnaliser le *blocklist*, chacun des trois composants a besoin de son propre élément. Avec des choix limités, Veerle a opté pour un élément **em** afin d'y mettre la description, et pour la date, un élément **span** et des balises **br** pour effectuer un retour à la ligne.

Veerle n'est certainement pas la seule à choisir cette approche. J'espère que **Elliot Jay Stocks** ne me tiendra pas rigueur d'utiliser la rubrique *Recent Reads* de son site comme un autre exemple. En effet, sur tous les sites où j'ai pu trouver une liste similaire, les auteurs ont dû employer des balises **strong**, **em**, **span**, **small** voire même **q** pour marquer le contenu de leurs liens.

## Quel est le problème ?

Coder la liste de cette manière soulève quelques interrogations. Faut-il vraiment souligner la description par rapport au reste du contenu du lien ? Y a-t-il une quelconque relation entre le titre de l'élément et sa description ? Qu'en est-il de la date ?

Deuxièmement, appliquer un style à la liste a été rendu plus difficile par le type inline des éléments. Des sauts de ligne ont dû être utilisés pour séparer ce qui aurait été une ligne continue de texte avec le CSS désactivé (sans les avantages du **display: block**).

Troisièmement, utiliser un élément comme la balise **em** juste pour *contenir* la description, empêche son utilisation pour du texte qui devrait être mis en valeur. Il en va de même pour les liens. Comment faire si vous souhaitez un lien supplémentaire dans la description ? Vous ne pouvez pas avoir un lien imbriqué.

## Un regard nouveau sur ces éléments

Quand je développais mon **site**, j'avais prévu d'y intégrer une liste de **Quicklinks** (liens rapides), c'est-à-dire un ensemble d'éléments externes à mon site renvoyant vers des sujets intéressants sur le Web, essentiellement le même que celui de la rubrique **Approved** de Veerle. J'ai pensé qu'il devait avoir une meilleure façon de construire une liste avec de larges zones cliquables, et donc je me suis mis à faire des tests.

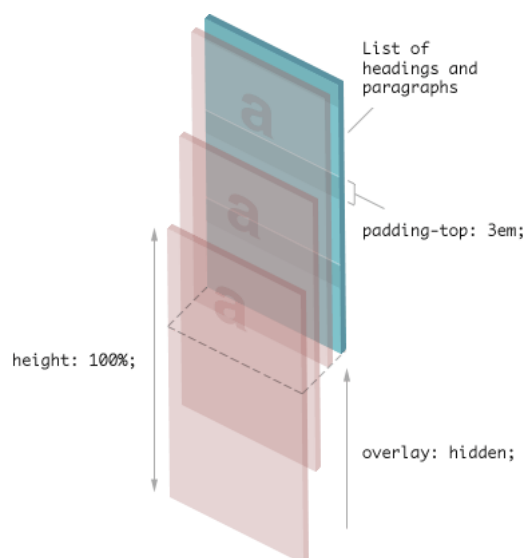
Pour démarrer, j'ai retranscrit la manière dont je voulais voir s'afficher les balises :

```
<h2>Liens Rapides</h2>
<ul>
  <li>
    <h3><a href="#">Titre du lien rapide</a></h3>
    <p>Une courte description du lien</p>
    <p>1er Janvier</p>
  </li>
</ul>
```

L'astuce de cette technique est que les liens se trouvent uniquement sur les titres. Les éléments **a** deviennent ensuite des éléments de type block (*display: block;*) qui recouvrent les éléments qui les suivent à la même manière qu'un calque transparent, qui rend l'ensemble cliquable.

Je tiens à souligner que tout le texte couvert par la balise étendue **a** ne pourra pas être sélectionné. Si cela vous pose un gros problème alors cette technique ne correspond peut-être pas à ce que vous recherchez.

Pour que le lien soit en haut du contenu qui lui est consécutif sans tout remettre en bas, il faut le retirer du flux du document (positionnement absolu) et lui attribuer un z-index assez important.



Le plus gros obstacle à surmonter est de prendre en compte un contenu variable ou l'augmentation de la taille du texte. Il serait facile de donner à chaque balise **a** une hauteur fixe et de la laisser telle quelle, mais elle ne a) couvrirait pas l'ensemble du contenu ou b) limiterait la quantité de texte qu'un lien pourrait avoir. Pour surmonter ce problème, la liste en elle-même doit avoir une position relative et chaque élément **a** une hauteur de 100%. Cela permet à chaque lien de s'étendre sur toute la hauteur de la liste. Les éléments qui apparaissent plus bas dans le code source ont à la base un z-index plus important, de ce fait chaque lien recouvre convenablement le précédent, ce qui coupe la zone cliquable depuis son origine (le haut du titre étant lié) jusqu'au point où le lien suivant débute (le haut du prochain titre).

En touche finale, on peut appliquer à la liste l'instruction " *overlay: hidden;* ". Attribuer à chaque lien une hauteur de 100% fait que ces derniers surplombent la liste qui doit se terminer à la hauteur des cases précédentes, cela donne une liste de 200% par rapport à la hauteur de base. Mettre la propriété *overlay* de liste à *hidden* découpe l'ensemble et la rétablit à la bonne hauteur. (Si vous avez du mal à assimiler tout cela, ne vous inquiétez pas. Reprenez le code de la page de **démonstration** et supprimez l'instruction *overflow: hidden;* pour voir ce que je veux dire).

Une des caractéristiques de la technique dont je ne me satisfais pas est l'obligation d'attribuer une certaine hauteur au titre du lien étant donné qu'il a été supprimé du flux du document. Pour y remédier, un padding d'environ 3em doit être ajouté en haut de l'élément qui suit immédiatement le titre. On fait cela en appliquant une classe à l'élément liste si son titre est susceptible de s'étendre sur deux ou plusieurs lignes et on applique un style à l'élément qui le suit de la manière suivante:

```
.longtitle h3 + p
{
    padding-top: 5em;
}

.verylongtitle h3 + p
{
    padding-top: 7em;
}
```

Si vous pensez qu'il y a une meilleure façon de surmonter cette difficulté, n'hésitez pas à manipuler le code et à me le faire savoir.

Eh bien, nous y sommes. La structure de base derrière cette technique est terminée. Tout ce qui nous reste à faire est de lui donner un style agréable.

## Astuces pour styliser votre Blocklist

- Utilisez le sélecteur de parenté pour cibler un élément consécutif à un autre (cf le deuxième paragraphe dans chaque élément de la liste : **p + p { color: red; }** ou, comme nous avons pu l'utiliser, la 'description' dans chaque lien : **h3 + p { padding-top: 3em; }**). Ces sélecteurs de parenté sont puissants, jusqu'à présent ils sont peu utilisés du fait de leur compatibilité plus que limitée par les navigateurs (je compte sur vous Microsoft).
- Pour appliquer un style à un lien lorsqu'il est survolé il faut appliquer la pseudo-class **:hover** à l'élément de la liste (**li**), pas au **a** lui-même. Par exemple, **li:hover { background-color: green; }**. Si vous appliquez **:hover** au **a**, vous ne serez pas en mesure de changer l'arrière-plan par crainte de masquer tout ce qui se trouve en dessous, et le style n'affectera pas le reste du contenu du lien.
- Si vous envisagez de mettre des liens supplémentaires dans les éléments regroupés dans **a** (un des avantages de cette technique), stylisez-les pour qu'ils apparaissent comme du texte normal étant donné qu'ils **ne seront pas cliquables**. N'importe quel utilisateur qui lira votre contenu avec le CSS désactivé ou dans un lecteur de flux RSS continuera à bénéficier de leur mise en forme mais je n'ai trouvé aucun moyen pour qu'ils restent cliquables par l'intermédiaire du lien qui les recouvre. Encore une fois, toutes suggestions relatives à une solution sur ce problème serait grandement appréciée dans vos commentaires.
- Amusez-vous avec cette technique ! Elle vous donne bien plus de flexibilité comparée à l'ancienne méthode et vous laisse un balisage propre pour styliser le tout à votre guise.

## Le Bon, le Méchant et IE

Eh bien, voici un petit tour d'horizon des avantages et inconvénients de cette technique :

	Avantages	Inconvénients
<b>Ancienne Méthode</b>	Grande zone cliquable sous IE6	Difficile à styliser Oblige à utiliser les retours à la ligne Exploite de façon incorrecte des éléments qui pourraient être mieux utilisés
<b>Nouvelle Méthode</b>	Propre, balisage sémantique Ne vous restreint pas à une utilisation des éléments inline Pas de balisage superflu Peut contenir des liens supplémentaires	La hauteur approximative du titre doit être connue A part le titre, le texte n'est pas sélectionnable Petite zone cliquable sous IE6

Comme vous pouvez le constater, c'est loin d'être parfait mais les avantages sont notables en comparaison de la méthode traditionnelle. Votre *blocklist* ne fonctionnera pas aussi bien sous IE6. Cependant n'ayez crainte, tel quel (en utilisant le code de la [démonstration](#)) le *blocklist* devrait s'afficher parfaitement, avec seulement des zones cliquables réduites à la taille de leurs titres.

L'un des avantages liés à l'utilisation de sélecteurs de parenté pour styliser le *blocklist* est qu'IE6 ignore la plupart des choses sur lesquelles il s'arrête habituellement. Bien sûr, les résultats peuvent varier si vous personnalisez le code en fonction de vos besoins, mais avec une ligne ou deux d'instructions CSS spécifiques cela devrait suffire à IE pour résoudre n'importe quel problème. Par défaut, IE7 le gère plutôt bien.

Et voilà ! Encore une fois, je tiens à remercier [Veerle](#) pour m'avoir permis d'utiliser son site à titre d'exemple. Assurez-vous d'avoir regardé la démonstration pour voir la technique en action. Elle inclut deux *blocklists* : l'un dans le même style que mes Quicklinks et l'autre, qui reprend à titre de comparaison la toute récente technique de Veerle dans sa rubrique Approved. Cependant, gardez à l'esprit que n'importe quel CSS fourni au-delà de la structure de base est seulement à titre de démonstration et ne doit pas être recopié aveuglément. J'espère que les styles que j'ai pu inclure vous donneront de quoi démarrer les vôtres.

Si vous avez un quelconque avis sur cette technique, positif ou négatif, n'hésitez pas à me le faire savoir.

## Remerciements

Tous mes remerciements à [RideKick](#) pour sa relecture.

1 :

Cet article a été publié à l'origine en version anglaise à l'adresse suivante : **[Building a Better Blocklist](#)**

2 :

Article traduit par l'équipe de traduction de la rédaction de developpez.com