

Un avant goût du HTML 5

par Lachlan Hunt ([A List Apart Magazine](#)) Traduit par Sylvain Jorge Do Marco

Date de publication : 16/10/2008

Le web est en constante évolution. De nouveaux sites innovants sont créés tous les jours, repoussant les limites du HTML dans toutes les directions. Le HTML 4 est présent partout depuis près d'une décennie maintenant et les éditeurs cherchant de nouvelles techniques pour fournir des fonctionnalités améliorées sont limités par les contraintes du langage et des navigateurs.

Pour donner aux auteurs plus de flexibilité et d'interopérabilité et rendre les sites web et applications plus interactifs et passionnants, le HTML 5 introduit et améliore un large éventail de fonctionnalités, comprenant les contrôles de formulaires, les APIs, le multimédia, la structure et la sémantique.

Le travail sur le HTML 5, commencé en 2004, est actuellement mené en collaboration par le **W3C HTML WG** et le **WHATWG**. Un grand nombre d'acteurs clés participent à l'effort du W3C dont des représentants des éditeurs des quatre principaux navigateurs : Apple, Mozilla, Opera et Microsoft ; et un éventail d'autres organisations et individus avec de nombreux intérêts et compétences différents.

Notez que la **spécification** est *toujours en écriture* et très loin d'être achevée. En tant que tel, il est possible que toute fonctionnalité abordée dans cet article change dans le futur.

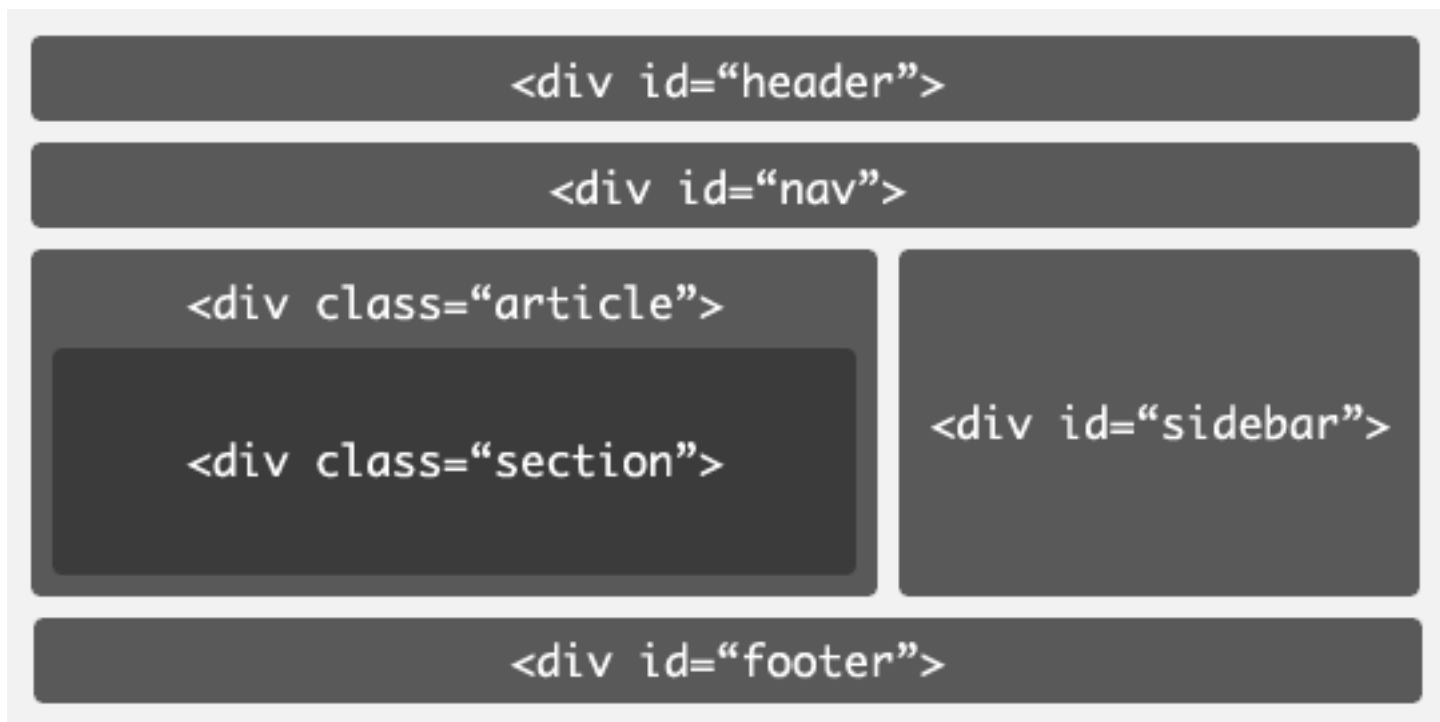
Cet article a pour objectif de fournir une brève introduction à certaines des fonctionnalités principales telles qu'elles sont dans l'ébauche actuelle.

La structure.....	3
Vidéo et Audio.....	7
Représentation du document.....	9
Comment contribuer.....	11
Remerciements.....	12

(1)

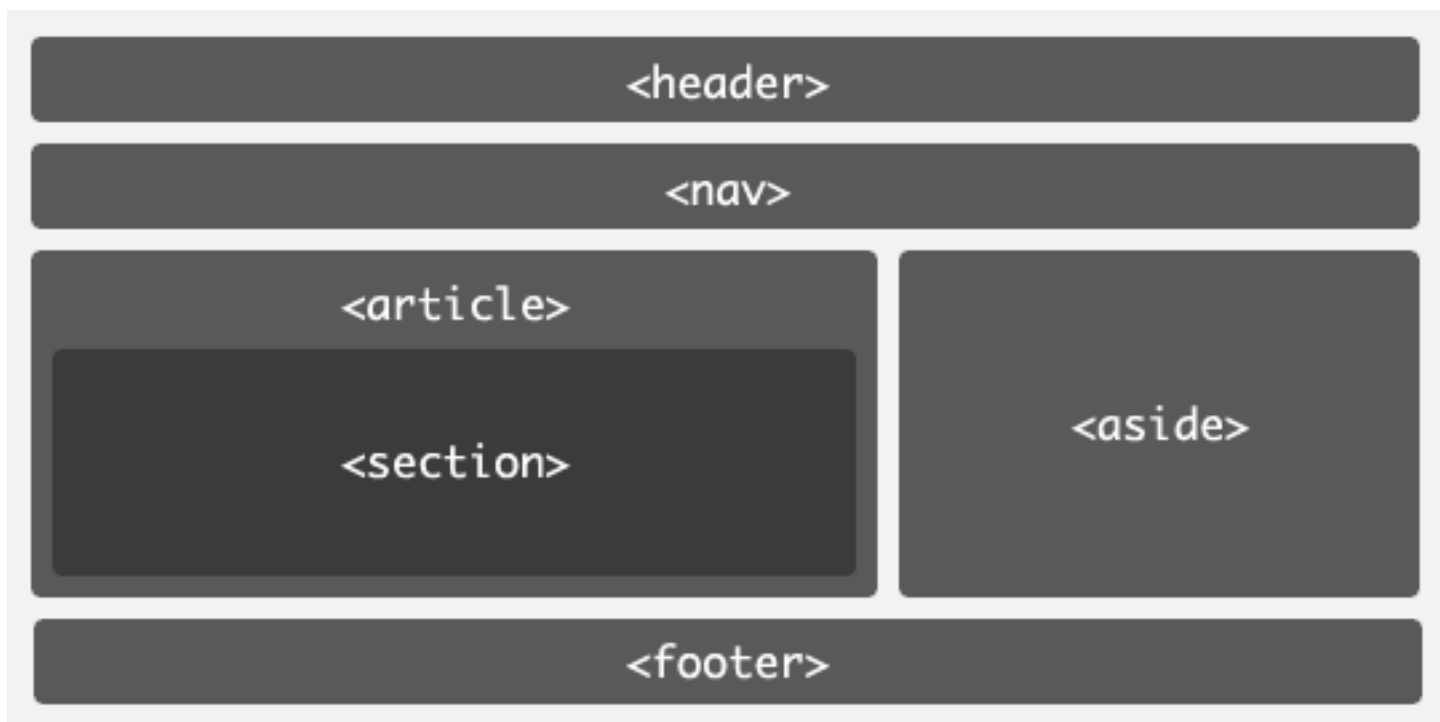
La structure

Le HTML 5 introduit une série complète de nouveaux éléments qui font qu'il est beaucoup plus simple de structurer les pages. La plupart des pages HTML 4 contiennent une diversité de structures identiques, comme les en-têtes (*header*), les bas de page (*footer*) et les colonnes. Aujourd'hui, il est relativement fréquent de les baliser par des éléments *div*, leur attribuant à chacun un identifiant ou une classe descriptive.



Ce diagramme illustre une présentation typique à deux colonnes, balisée à l'aide de *div* ayant des attributs *id* et *class*. Elle contient un en-tête, un pied de page, et une barre de navigation horizontale sous l'en-tête. Le contenu principal comprend un article et une barre latérale à sa droite.

L'utilisation d'éléments *div* est due principalement aux versions actuelles du HTML 4 qui n'ont pas la sémantique nécessaire pour décrire ces parties plus spécifiquement. Le HTML 5 s'attèle à ce problème en introduisant de nouveaux éléments pour représenter chacune de ces différentes sections.



Les éléments div peuvent être remplacés par de nouveaux éléments : *header*, *nav*, *section*, *article*, *aside* et *footer*.

Le code de ce document pourrait ressembler à celui-ci :

```

<body>
  <header>...</header>
  <nav>...</nav>
  <article>
    <section>...</section>
  </article>
  <aside>...</aside>
  <footer>...</footer>
</body>

```

Il y a plusieurs avantages à utiliser ces éléments. Utilisés en conjonction avec les éléments de titre (*h1* à *h6*), ils fournissent tous un moyen de baliser les sections imbriquées avec des niveaux de titre, au delà des six niveaux possibles avec les versions précédentes du HTML. La spécification comprend **un algorithme détaillé de génération de sommaire** qui prend en compte la structure de ces éléments et reste compatible avec les versions précédentes. Il peut être utilisé à la fois par les outils d'édition et les navigateurs pour générer des tables des matières et ainsi aider les utilisateurs à naviguer dans le document.

Par exemple, la structure de balises suivante utilise des éléments *section* et *h1* imbriqués :

```

<section>
  <h1>Niveau 1</h1>
  <section>
    <h1>Niveau 2</h1>
    <section>
      <h1>Niveau 3</h1>
    </section>
  </section>
</section>

```

Notez que pour une meilleure compatibilité avec les navigateurs actuels, il est aussi possible d'utiliser de façon appropriée les autres éléments de titre (*h2* à *h6*) à la place des éléments *h1*.

En identifiant le rôle des sections dans la page utilisant des éléments spécifiques de séparation, une technologie d'assistance peut aider l'utilisateur à naviguer plus facilement sur la page. Par exemple, ils peuvent facilement sauter la section de navigation ou rapidement aller d'un article au suivant sans que les auteurs aient besoin de fournir des liens spécifiques. Les auteurs en profitent également puisque remplacer la plupart des *div* du document par un des quelques éléments distincts peut aider à rendre le code source plus clair et plus simple pour l'auteur.

L'élément *header* représente l'en-tête d'une section. Les en-têtes peuvent contenir plus qu'un simple titre de section. Par exemple, il serait raisonnable d'inclure dans le header des sous-titres, une information sur l'historique de version ou des légendes.

```
<header>
  <h1>A Preview of HTML 5</h1>
  <p class="byline">By Lachlan Hunt</p>
</header>

<header>
  <h1>Blog exemple</h1>
  <h2>Insérer une ligne de citation ici.</h2>
</header>
```

L'élément *footer* représente le bas de la section à laquelle il s'applique. Un *piéd* contient typiquement une information sur sa section comme son auteur, des liens vers des documents liés, les données de copyright et autres données du même type.

```
<footer>© 2007 Exemple Inc.</footer>
```

L'élément *nav* représente une section de liens de navigation. Il convient à la fois pour la navigation dans le site ou une table des matières.

```
<nav>
  <ul>
    <li><a href="/">Accueil</a></li>
    <li><a href="/produits">Produits</a></li>
    <li><a href="/services">Services</a></li>
    <li><a href="/apropos">A propos</a></li>
  </ul>
</nav>
```

L'élément *aside* est destiné au contenu qui est indirectement lié au contenu qui l'entoure et est typiquement utile pour baliser des barres latérales.

```
<aside>
  <h1>Archives</h1>
  <ul>
    <li><a href="/2007/09/">Septembre 2007</a></li>
    <li><a href="/2007/08/">Août 2007</a></li>
    <li><a href="/2007/07/">Juillet 2007</a></li>
  </ul>
</aside>
```

L'élément *section* représente une section générique d'un document ou d'une application, comme un chapitre par exemple.

```
<section>
  <h1>Chapitre 1: La Période</h1>
  <p>C'était le meilleur des temps, c'était le plus mauvais des temps,
    c'était l'âge de la sagesse, c'était l'âge de la sottise,
    c'était l'époque de la croyance, c'était l'époque de l'incrédulité,
    c'était la saison de la lumière, c'était la saison de l'obscurité,
  ...</p>
</section>
```

(extrait de '*A Tale of Two Cities*')
(source: [http://www.w3.org/html/wg/drafts/html5/draft-html5-2/sections.html](#))

L'élément *article* représente une section indépendante d'un document, d'une page ou d'un site. Il convient pour du contenu comme des *nouvelles* ou des articles de blog, des messages de forum ou des commentaires individuels.

```
<article id="commentaire-2">
  <header>
    <h4>
      <a href="#commentaire-2" rel="bookmark">Commentaire #2</a>
      par <a href="http://example.com/">Jack O'Niell</a>
    </h4>
    <p>
      <time datetime="2007-08-29T13:58Z">29 Août 2007 à 13:58</time>
    </p>
  </header>
  <p>Voici un autre article génial !</p>
</article>
```

Vidéo et Audio

Ces dernières années, la vidéo et l'audio sur le web sont devenus de plus en plus viables et des sites comme **YouTube**, **Viddler**, **Revver**, **MySpace** et des douzaines d'autres ont rendu plus simple pour tout le monde la publication de vidéo et d'audio. Cependant, comme le HTML n'offre pas par lui-même les moyens nécessaires pour intégrer et contrôler avec succès le multimédia, de nombreux sites dépendent de Flash pour fournir cette fonctionnalité. Bien qu'il soit possible d'intégrer du multimédia en utilisant divers plugins (comme QuickTime, Windows Media, etc.), Flash est actuellement le seul plugin largement déployé qui fournit une solution compatible multi-navigateur avec les APIs appréciées par les développeurs.

Comme cela a été mis en évidence par les divers lecteurs de médias basés sur Flash, les auteurs sont intéressés par le fait de fournir leurs propres interfaces utilisateurs qui, généralement, permettent aux utilisateurs de lire, mettre en pause, arrêter, chercher, et ajuster le volume. L'objectif est de fournir cette fonctionnalité dans les navigateurs en ajoutant un support natif de vidéos et audios embarqués et fournir des APIs DOM pour que les scripts contrôlent la lecture.

Les nouveaux éléments *video* et *audio* le rendent vraiment facile. La plupart des APIs sont communes aux deux éléments, les seules différences étant liées aux différences inhérentes entre des médias visuels et non visuels.

Opera et **WebKit** ont tous les deux publié des versions avec support partiel de l'élément *video*. Vous pouvez télécharger la **version expérimentale d'Opera** ou une récente **version quotidienne de WebKit** pour essayer ces exemples. Opera inclut le support d'Ogg Theora et WebKit supporte tous les formats supportés par QuickTime y compris les codecs tiers.

La manière la plus simple d'intégrer une vidéo est d'utiliser un élément *video* et de permettre au navigateur de fournir une interface utilisateur par défaut. L'attribut *controls* est un attribut booléen qui indique si l'auteur souhaite que cette interface utilisateur soit présente ou non par défaut.

```
<video src="video.ogv" controls poster="poster.jpg" width="320" height="240">
  <a href="video.ogv">Télécharger le film</a>
</video>
```

L'attribut facultatif *poster* peut être utilisé pour spécifier une image qui sera affichée à la place de la vidéo avant qu'elle ne commence. Bien qu'il y ait quelques formats vidéo qui ont leur propre fonctionnalité *poster*, comme le MPEG-4, cela fournit une solution alternative qui peut fonctionner indépendamment du format vidéo.

Il est aussi simple d'intégrer de l'audio à une page en utilisant l'élément *audio*. La plupart des attributs sont communs aux éléments *video* et *audio*, bien que pour d'évidentes raisons, l'élément *audio* n'ai pas d'attributs *width*, *height* et *poster*.

```
<audio src="music.oga" controls>
  <a href="music.oga">Télécharger la chanson</a>
</audio>
```

Le HTML 5 fournit l'élément *source* pour spécifier des fichiers alternatifs vidéo et audio que le navigateur peut choisir en fonction de leur type ou des codecs supportés. L'attribut *media* peut être utilisé pour spécifier une demande de sélection basée sur les limitations de l'appareil et l'attribut *type* pour spécifier le type et les codecs du média. Notez que lorsque les éléments *source* sont utilisés, l'attribut *src* doit être omis de l'élément parent vidéo ou audio ou les alternatives données par les éléments *source* seront ignorés.

```
<video poster="poster.jpg">
  <source src="video.3gp" type="video/3gp" media="handheld">
  <source src="video.ogv" type="video/ogg; codecs=theora, vorbis">
```

```
<source src="video.mp4" type="video/mp4">
</video>
<audio>
  <source src="music.oga" type="audio/ogg">
  <source src="music.mp3" type="audio/mpeg">
</audio>
```

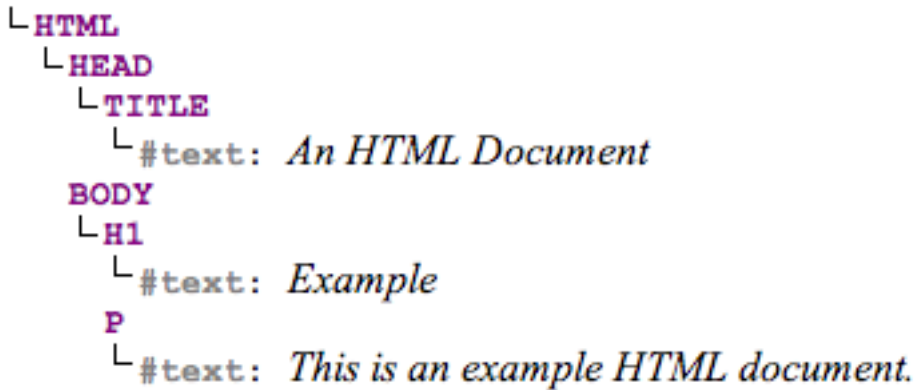
Pour les auteurs qui souhaitent un peu plus de contrôle sur l'interface utilisateur afin de la faire correspondre au design général de la page web, l'API, riche, fournit plusieurs méthodes et événements pour laisser aux scripts le contrôle de la lecture du média. Les méthodes les plus simples à utiliser sont *play()*, *pause()* et positionner *currentTime* pour revenir au début. L'exemple qui suit illustre leur utilisation.

```
<video src="video.ogg" id="video"></video>
<script>
  var video = document.getElementById("video");
</script>
<p>
  <button type="button" onclick="video.play();">Lecture</button>
  <button type="button" onclick="video.pause();">Pause</button>
  <button type="button" onclick="video.currentTime = 0;"><< Rembobiner</button>
</p>
```

Il y a de nombreux autres attributs et APIs disponibles pour les éléments vidéo et audio qui n'ont pas été présentés ici. Pour plus d'informations, vous pouvez consulter la [spécification actuelle du projet](#).

Représentation du document

Contrairement aux versions précédentes du HTML et du XHTML, qui sont définies avec les termes de leur syntaxe, le HTML5 est défini avec les termes du *Document Object Model* (DOM), l'arbre de représentation utilisé en interne par les navigateurs pour représenter le document. Par exemple, considérons un document très simple composé d'un en-tête, d'un titre et d'un paragraphe. L'arbre DOM pourrait ressembler à quelque chose comme ça :



L'arbre DOM comprend un élément title dans l'en-tête et des éléments h1 et p dans le corps.

L'avantage de définir le HTML 5 avec les termes du DOM est que le langage lui-même peut être défini indépendamment de la syntaxe. Il y a essentiellement deux syntaxes qui peuvent être utilisées pour représenter des documents HTML : la sérialisation HTML (connue en tant qu'HTML 5) et la sérialisation XML (connue en tant qu'XHTML 5).

La sérialisation HTML se rapporte à la syntaxe inspirée par la syntaxe SGML des versions précédentes du HTML, mais définie pour être plus compatible avec la manière dont les navigateurs traitent réellement le HTML en pratique.

```

<!DOCTYPE html>
<html>
  <head>
    <title>Un document HTML</title>
  </head>
  <body>
    <h1>Exemple</h1>
    <p>Voici un exemple de document HTML.
  </body>
</html>
  
```

Notez que comme dans les précédentes versions du HTML certaines balises sont optionnelles et sont automatiquement implicites.

La sérialisation XML se rapporte à la syntaxe utilisant le XML 1.0 et les espaces de noms, tout comme le XHTML 1.0.

```

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Un document HTML</title>
  </head>
  <body>
    <h1>Exemple</h1>
    <p>Voici un exemple de document HTML.</p>
  </body>
</html>
  
```

Mises à part les différences d'espaces blancs et la présence de l'attribut *xmlns*, ces deux exemples sont équivalents.

Les navigateurs utilisent le type MIME pour distinguer les deux. Tous documents présentés comme *text/html* doivent être conformes aux conditions de la sérialisation HTML et tous documents présentés avec un type MIME XML comme *application/xhtml+xml* doivent se conformer aux conditions de la sérialisation XML.

Les auteurs devraient faire un choix éclairé sur la sérialisation à utiliser, qui peut dépendre du nombre de différents facteurs. Les auteurs ne devraient pas être forcés sans condition à utiliser l'un ou l'autre ; chacun est optimisé pour différentes situations.

Les bénéfices de l'utilisation du HTML

- Compatibilité descendante avec les navigateurs existants
- Les auteurs sont déjà familiers avec la syntaxe
- La syntaxe légère et indulgente signifie qu'il n'y aura pas d'"**écran jaune de la mort**" anti-utilisateur si une erreur se glisse accidentellement
- Une syntaxe sténographique pratique, i.e. les auteurs peuvent omettre certaines balises et valeurs d'attributs

Les bénéfices de l'utilisation du XHTML

- Une syntaxe XML stricte encourage les auteurs à écrire des balises bien formées, que certains auteurs peuvent trouver plus facile à maintenir
- S'intègre directement avec d'autres vocabulaires XML, comme SVG et MathML
- Permet l'utilisation d'un parseur XML, que certains auteurs utilisent comme une part de leurs procédés d'édition et/ou de publication

Comment contribuer

Le travail sur le HTML 5 avance rapidement, pourtant il est encore prévu qu'il continue plusieurs années. En raison de l'exigence de produire des cas de test et de réaliser des implémentations interopérables, la fin des travaux est actuellement estimée à environ dix ou quinze ans. Pendant ce processus, les retours d'un large éventail de personnes comprenant, entre autres, les concepteurs et développeurs web, les vendeurs de CMS et d'outils d'édition, et les vendeurs de navigateurs est primordial pour assurer son succès. Tout le monde n'est pas seulement bienvenu, mais activement encouragé à contribuer aux remarques sur le HTML 5.

En plus de la spécification, il y a plusieurs autres efforts apparentés destinés à aider les personnes à mieux comprendre le travail.

- **Les différences depuis le HTML 4** décrivent les modifications qui sont survenues depuis la dernière version du HTML.
- **Les principes du design HTML** traitent des principes utilisés pour aider à prendre des décisions et vous aideront à comprendre la logique derrière la plupart des décisions actuelles de design.
- **Le guide du développeur Web en HTML 5**, qui a commencé récemment, est écrit pour aider les designers et les développeurs à comprendre tout ce qu'ils ont à savoir pour écrire des documents conformes au HTML 5, fournit des conseils et décrit les bonnes pratiques.

Il y a de nombreux lieux où vous pouvez contribuer. Vous pouvez rejoindre le **groupe de travail du W3C (W3C's HTML WG)** et vous inscrire / contribuer aux **listes de diffusion du groupe de travail HTML** ou au **wiki**. Vous pouvez aussi vous inscrire et contribuer aux **listes de diffusion du WHATWG**, poster dans le **forum WHATWG**, poster des commentaires ou écrire des articles dans le **blog WHATWG**.

Remerciements

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

1 :

Traduction de l'article **A Preview of HTML 5** de Lachlan Hunt paru sur A List Apart le 4 décembre 2007.
Translated with the permission of **A List Apart Magazine** and the author.