

# HTML5 et CSS3

## Chap 01

### Avant-propos

**L'écriture de ce livre sur le Html5 et le CSS3 a été une entreprise très motivante. Le Html n'avait pas connu d'évolution majeure depuis 1998 avec le Html 4.0. Une éternité à l'échelle du développement du Web et d'Internet ! Il est important d'évaluer les évolutions et apports de cette nouvelle version du langage qui est à la base de cet outil devenu quotidien qu'est le Web. De leur côté, les feuilles de style CSS3 constituent une réelle avancée au niveau du graphisme des pages avec, par exemple, les bords arrondis, la transparence des images, les ombres ou la typographie.**

**Cet ouvrage sur le Html5 et les CSS3 se veut avant tout un ouvrage structuré et formatif plutôt qu'une exploration exhaustive des nouveautés apportées par ces deux technologies. Il vise deux publics bien distincts :**

**Les débutants pourront appréhender les bases du langage source des pages de la toile au fil des chapitres grâce à de nombreux exemples complets, illustrés à chaque fois par des captures d'écran et la possibilité de les visionner sur leur écran, via l'espace de téléchargement consacré à cet ouvrage. Ils pourront ainsi entamer leur apprentissage de la conception de pages Web par la dernière évolution du Html qui est d'ores et déjà opérationnelle dans de nombreux navigateurs de la dernière génération et qui sera le standard des années futures.**

**Ce livre s'adresse aussi aux concepteurs familiarisés au Html et qui souhaitent découvrir les apports et nouveautés du Html5 et des CSS3. Les différences par rapport au Html 4.0, parfois subtiles mais déterminantes, sont mises en évidence tout au long de l'ouvrage. Ils pourront aussi découvrir les réelles nouveautés du Html5 avec les chapitres dédiés aux nouveaux champs de formulaires, au dessin 2D, aux nouvelles balises audio et vidéo et à l'entrée du JavaScript avec, par exemple, la géolocalisation de l'utilisateur. L'important chapitre sur les CSS3 les convaincra de la révolution qu'elles vont apporter dans le design de la page, les débarrassant du passage obligé par des programmes graphiques.**

**Cet ouvrage marque également un tournant dans l'apprentissage du code source des pages de la toile. En effet, le Html5 fait table rase de tout aspect de présentation pour se consacrer exclusivement à la structure du document. Il n'est donc plus possible en Html5 d'aligner du texte ou autres éléments ou de dessiner des bordures de tableaux. Terminé aussi l'ajout de couleur au texte ou en arrière-plan. Le Html5 se décline désormais de façon austère, en noir et blanc ! Tous les éléments de présentation doivent obligatoirement être repris par les feuilles de style. Il n'est plus possible de concevoir un apprentissage de manière traditionnelle avec une partie exclusivement consacrée au Html et une seconde partie distincte pour les feuilles de style. Ces deux éléments sont maintenant intimement liés. Ainsi, il nous a semblé utile, voire indispensable, d'introduire dès l'apprentissage du Html des éléments basiques relatifs aux feuilles de style. Cela permettra à l'apprenant de concevoir ses premières applications d'une manière plus réaliste et plus attrayante.**

**Entrons plus profondément dans le détail des chapitres. La partie Html débute par les éléments basiques et traditionnels de l'apprentissage du code source. Le Html5 est en premier lieu une évolution du Html 4.0. Il reste donc un langage de balises. Les premiers éléments de l'apprentissage progressif concernent le texte, la structuration du document avec les titres et les listes, les tableaux, les images et leur insertion, sans oublier ce qui fait l'essence du langage Html c'est-à-dire les liens. Une série de chapitres aborde ensuite les nouveautés introduites par le Html5.**

**Les nouvelles balises sémantiques qui permettent d'organiser le code source des applications afin d'en assurer une meilleure lisibilité.**

**Il faut bien avouer que les formulaires traditionnels comme la ligne ou la zone de texte, les boutons radio, à choix multiples ou autres boutons de commandes entraînent une lassitude certaine dans l'aspect visuel de la page. Les apports du Html5 vont révolutionner les formulaires avec des lignes de texte avec suggestions, des calendriers, des compteurs numériques, des curseurs, une validation sans passer par le JavaScript et bien d'autres surprises à découvrir dans le chapitre qui leur est consacré.**

**Les nouvelles balises audio et vidéo permettent désormais de lire de façon native les fichiers multimédia sans devoir passer par des plugins divers, souvent sources de complications.**

**Le dessin 2D ouvre de nouvelles perspectives car il est désormais possible, par exemple, d'ajouter des graphiques directement**

**par le code, sans passer par des captures d'écran de programmes comme Excel.**

**Pour terminer, il fallait souligner que le Html5 marque aussi l'introduction du JavaScript comme un acteur privilégié dans la conception des applications Web. Ce dernier point est illustré par la géolocalisation de l'internaute.**

**Poursuivons avec la partie dédiée aux feuilles de style CSS. Dans un apprentissage progressif et structuré, il n'est pas possible d'ignorer les désormais élémentaires CSS1 et CSS2. Plusieurs chapitres sont ainsi consacrés aux feuilles de style relatives à la police de caractères, au texte, aux listes et tableaux, à la couleur, aux marges intérieures et extérieures, aux pseudo-classes et aux feuilles de style relatives à l'affichage ou à l'impression. Cette partie se termine par les spécifications CSS3 qui vont profondément modifier l'aspect visuel de la navigation sur la toile. Ce chapitre ne manquera pas d'intéresser ceux qui, de près ou de loin, prennent en charge l'aspect graphique des sites. Parmi les apports de ces feuilles de style CSS3, citons :**

**Les bords arrondis qui permettent de sortir de la présentation rectangulaire et angulaire des divisions.**

**Les ombres que l'on peut apporter au texte ou aux images sans passer par une application graphique.**

**Les polices que le concepteur pourra enfin choisir librement, marquant ainsi l'introduction tant attendue de la typographie dans la conception des pages web.**

**La présentation d'un contenu textuel en colonnes.**

**Les bordures imagées.**

**Les arrière-plans multiples.**

**Les dégradés de couleurs.**

**La possibilité de jouer sur l'opacité ou la transparence d'éléments comme les images.**

**À n'en pas douter, cet ouvrage est dense mais présenté dans un langage que nous avons voulu clair, formatif et pédagogique.**

**En tant qu'auteur, je dois avouer que j'ai pris un énorme plaisir à écrire ce livre sur le Html5 et les CSS3. Je fais le souhait que ce plaisir soit partagé avec les lecteurs.**

## Bref historique du Html5

**Nous débutons cet historique en 1998 avec la finalisation de la spécification Html 4.0. Mis à part une légère révision en 1999 avec le Html 4.01, plus rien de neuf à l'horizon de ce langage premier du Web. Le W3C (*World Wide Web Consortium*), l'organisme qui gère les standards d'écriture sur la toile, n'a-t-il pas annoncé à l'époque que le Html dans sa version 4.0 ne connaîtrait à l'avenir plus de développement, en bref que le Html était mort ?**

**Le W3C avait aussi édité en 1998 le XML (*eXtensible Markup Language* ou, en français, le langage extensible de balises) pour gérer de façon structurée des données de type texte, l'aspect extensible permettant à l'utilisateur de définir son propre langage avec ses éléments distincts. Le W3C recommandait à l'époque fortement le XML pour exprimer des langages de balisages spécifiques.**

**Une des premières applications concrètes du XML fut la reformulation du Html 4.0 selon la syntaxe stricte et formaliste du XML. Le Xhtml 1.0 apparût ainsi en 2000. Si les développeurs ont unanimement salué la rigueur que le Xhtml apportait à l'écriture du code, il faut bien constater que la migration vers le Xhtml n'a pas remporté le succès souhaité et que bon nombre de concepteurs sont restés fidèles au Html 4.0 transitional, plus souple.**

**Très rapidement, le W3C mit en chantier l'ambitieux projet du Xhtml 2.0 qui devait faire table rase du passé en matière de publication sur la toile. Cette position induisait que le Xhtml 2.0 ne serait pas rétrocompatible avec ce qui existait à l'époque et donc avec le Html.**

**Cette absence de rétrocompatibilité a engendré un mécontentement certain auprès des firmes qui développaient les navigateurs.**

**Ce qui eut pour conséquence la création d'un groupe dissident, le WHATWG (*Web Hypertext Application Technology Working Group*). Ce groupe de travail se présente notamment comme une réponse à la lenteur du développement des standards par le W3C et au caractère trop fermé de son processus interne d'élaboration de spécifications. Il se compose à l'origine de représentants de firmes comme Mozilla, Opera et Apple dont les navigateurs Firefox, Opera et Safari sont bien connus. Ils furent rejoints ensuite par Google. Leur position était diamétralement opposée à celle du Xhtml 2.0 en travaillant de façon pragmatique sur base des implémentations actuelles et donc du Html 4.0. Leurs premières réalisations portaient sur les Web Forms 2.0 pour**

**renouveler les formulaires et le nouveau concept de publication sur le web, Web Apps 1.0 soit des applications Web.**

**Parallèlement le Hhtml 2.0, même si un brouillon (working draft) paraît en juillet 2006, connaît un développement particulièrement laborieux et la rumeur rapporte de nombreuses dissensions au sein de ce groupe de travail. En outre, les différents navigateurs précités boudaient délibérément le Html 2.0.**

**Il apparaissait clairement que le Xhtml 2.0 connaissait de gros problèmes. Sir Tim Berners-Lee, inventeur du Web et président du W3C, décida fin 2006, de rouvrir un groupe de travail sur le Html qui, de façon pragmatique, devait reprendre la suite des travaux du WHATWG. Ainsi, il y eut pendant tout un temps, le groupe de travail sur le Xhtml 2.0 et celui sur le Html5. Ce dernier publia le 22 janvier 2008, un premier brouillon de travail qui a connu de nombreuses évolutions depuis.**

**Mais les jours du Xhtml 2.0 étaient comptés. Fin 2009, le même Sir Tim Berners-Lee annonce la dissolution du groupe de travail sur le Xhtml 2.0 et l'abandon définitif de ce dernier.**

**On annonce pour 2012, la *candidate recommendation* du Html5 pour 2012 et, étant donné l'ampleur des nouveautés annoncées, la recommandation finale pour... 2022.**

**L'annonce du Html5 a fait grand bruit (un buzz dans le vocabulaire moderne). Cet intérêt s'est rapidement concrétisé à différents niveaux. Google a incorporé très rapidement, soit à partir de la version 5 de Chrome des éléments du Html5. Ses autres comparses du WHATWG également avec les versions récentes de Firefox, Safari et Opera. Même Microsoft, pourtant souvent à la traîne en matière d'adoption de nouveautés et de standards incorpore le Html5 dans la version 9 d'Internet Explorer. De leur côté, les smartphones adoptent également le standard Html5, que ce soit Apple avec son iPhone, RIM avec son BlackBerry, Google avec son Google Phone, etc. Est-ce encore une évolution ? N'est-ce pas une révolution ?**

**Parallèlement, une reformulation du Html5 selon les règles et la syntaxe du XML est en cours d'élaboration. Celle-ci porte le nom de Xhtml5.**

**De cette gestation mouvementée du Html5, on retiendra :**

**Que l'on ne parle plus de pages Web mais du concept plus large d'applications Web.**

**Que le Html5 est une évolution (importante certes) du Html 4.0.**

**Que le Html5 a été conçu directement par les navigateurs les plus innovants du marché comme Firefox, Safari Opéra et Google. Ce qui est la garantie d'une adoption rapide des standards du Html5.**

**Que l'on peut s'attendre avec le Html5 à de grands changements concernant les formulaires (voir les *Web Forms 2.0*).**

**Qu'après plus d'une décennie sans réelles nouveautés, le Html5 correspond à un réel besoin des concepteurs pour renouveler l'interface des applications Web.**

## Lignes de force du Html5

**Avant de parcourir les nouveautés du Html5, dégageons quelques lignes de force de cette nouvelle version du langage Html.**

**Le Html5 est une évolution du Html. Il en reprend les grands principes, quitte à en améliorer certains aspects. Il est conçu pour assurer une rétrocompatibilité avec ce qui a été fait jusqu'à présent, en termes de publication sur le Web.**

**Même si le codage est moins formaliste que le Xhtml5, l'exigence d'un code propre, respectueux des règles fondamentales du balisage continue d'être d'actualité.**

**Le principe de la séparation du contenu et de la présentation reste de mise et se voit même renforcé. L'abandon de toute une série de balises et d'attributs de présentation (point F du présent chapitre) en est la meilleure preuve. Les feuilles de style CSS sont des partenaires indissociables du Html5. Celui-ci espère qu'avec la montée en puissance des nouvelles feuilles de style CSS3, l'interface de l'utilisateur soit complètement transformée dans les années à venir.**

**La simplification du code et le désir d'éviter toutes complications souvent inutiles sont deux principes suivis dans l'élaboration du Html5. On retrouve cette simplification et ce pragmatisme par exemple, dans un nouveau doctype simplifié et un code Html dépouillé de tout aspect de présentation.**

**Une chasse aux plugins qui viennent encombrer le code et compliquer la tâche des concepteurs, par leur prise en charge**

**directe dans le navigateur. Les nouvelles balises audio et vidéo en sont une belle illustration.**

**La reconnaissance du JavaScript comme partenaire de la publication sur la toile. On pense ainsi aux scripts classiques pour vérifier l'encodage obligatoire ou la validité d'une adresse de courrier électronique dans les formulaires. Le Html5 évitera ces encodages redondants par la prise en charge de façon native par les navigateurs de ces fonctions. Autre exemple, le codage du dessin 2D et bientôt 3D (balise <canvas>) est beaucoup plus proche du JavaScript que d'un langage de balise.**

**Le Html5 devient également une plate-forme d'interfaces d'applications (APIs) intégrant des fonctionnalités complexes comme la géolocalisation, l'édition en ligne ou le glisser-déposer (*drag/drop*).**

## Les nouveautés du Html5

**Les nouveautés du Html5 et surtout du trinôme du concept "Html5 + CSS3 + applications JavaScript" sont nombreuses.**

### Html5

**Un nouveau doctype simplifié et unifié.**

**La suppression des balises et attributs de présentation.**

**De nouvelles balises sémantiques ou d'organisation.**

**De nouvelles balises audio et vidéo qui ne nécessitent plus l'appel à des plug-ins dédiés.**

**Le dessin 2D et bientôt 3D par la nouvelle balise <canvas>.**

**Une profusion de formulaires novateurs comme les curseurs ou les calendriers et la prise en charge de façon native par les navigateurs de la validation des données.**

**Etc.**

**L'objectif premier de cet ouvrage est bien entendu l'étude détaillée de l'héritage du Html 4.0 et des nouveautés du Html5. Il faut noter que ce dernier est déjà bien implanté dans les navigateurs de la dernière génération. L'implantation actuelle du Html5 permet déjà de le découvrir et de l'utiliser. Il faut**

cependant noter que l'intégration du Html5 n'est pas encore complète. De nouveaux apports sont encore attendus, particulièrement en ce qui concerne les formulaires.

### CSS3

**De nouveaux sélecteurs.**

**Les bords arrondis.**

**Les bordures imagées.**

**Les polices personnalisées permettant à la typographie de trouver enfin sa place dans les interfaces Web.**

**La présentation en plusieurs colonnes d'un texte.**

**Les ombres sur le texte comme sur les éléments.**

**Les dégradés de couleur.**

**Les arrière-plans multiples.**

**L'opacité ou la transparence.**

**Les transformations.**

**Les transitions (sans JavaScript).**

**Ces nouvelles spécifications CSS3 sont déjà disponibles dans les versions récentes des navigateurs et en particulier dans les navigateurs de notre étude (voir la section "Les navigateurs de notre étude" dans le présent chapitre). Il ne fait aucun doute que les CSS3 vont révolutionner la façon de concevoir et d'exploiter les interfaces Web dans les prochains mois ou les prochaines années.**

### Les applications JsAPI

**La géolocalisation qui permet de localiser (avec sa permission) l'utilisateur par ses coordonnées de longitude et latitude.**

**Des super cookies avec *Web Storage* qui permettra un stockage plus important de données dans le navigateur.**

**L'utilisation des applications Web hors connexion après la mise en cache des ressources nécessaires.**

**Les *Workers* qui peuvent exécuter des tâches de fond en parallèle du programme JavaScript principal dans un environnement totalement séparé de la page.**

**Les *Websockets* qui permettent d'établir une communication**

bidirectionnelle asynchrone entre le navigateur et le serveur.

Le glisser-déposer (*drag/drop*) en natif dans le navigateur.

L'attribut *ContentEditable* qui permet l'édition en ligne du contenu d'un élément. Il fait apparaître un éditeur WYSIWYG basique qui permet donc d'éditer directement le contenu dans la page. Toutes les modifications apportées par l'utilisateur pourront être traitées à la fin de l'édition.

Etc.

Ces applications dépassent largement le cadre de cet ouvrage dédié à l'apprentissage du HTML5. Certaines sont déjà implémentées dans les navigateurs récents, d'autres non. Nous aborderons cependant brièvement la géolocalisation dans un chapitre pour son côté spectaculaire et intrigant.

Un certain nombre de ces fonctions sont déjà reprises dans les cadres (framework) JavaScript comme Dojo et jQuery.

## Le HTML5 est un langage de balises

Digne successeur des versions précédentes du HTML et du HTML 4.0 en particulier, le HTML5 est toujours un langage de balises.

### 1. Les balises

Les balises, aussi appelées éléments, sont des commandes à l'intention du navigateur et saisies entre des signes inférieur à (<) et supérieur à (>). Ainsi une balise s'écrit <balise>.

En règle générale, à toute balise d'ouverture correspond une balise de fermeture qui marque la fin de la commande annoncée par la balise d'ouverture. La balise de fermeture reprend le même énoncé que la balise d'ouverture mais ce dernier est précédé d'une barre oblique (/). Ainsi à la balise d'ouverture <balise> correspondrait la balise de fermeture </balise>. La syntaxe d'une balise est alors :

```
<balise> ... </balise>
```

Pour comprendre le mode de fonctionnement des balises, je vous propose le texte suivant :

Il est **important** d'apprendre le langage *HTML5* !

Ceci peut se comprendre de cette façon :

écrire "Il est" de façon normale puisque rien n'est spécifié,  
ensuite écrire le mot (et uniquement ce mot) "important" et le  
mettre en gras,

repandre l'écriture normale pour "d'apprendre le langage",

écrire cette fois-ci en italique le mot "Html5"

et terminer par "!" en écriture normale.

Ce qui donne le résultat suivant :

**Il est** important *d'apprendre le langage* *Html5* !

Le langage Html n'est ni plus ni moins que cela. Chaque fois que l'on donne une instruction au navigateur, par exemple mettre un titre, commencer un tableau ou faire un lien vers une autre page, une balise de début est appliquée. La balise de fermeture signale au navigateur que la commande est terminée.

Une exception cependant, Html5 a hérité du Html des balises uniques, aussi appelées balises vides qui n'ont pas de balises de fermeture.

Par exemple, la balise image <img>.

Afin d'assurer une rétrocompatibilité avec le Xhtml, ces balises peuvent également s'écrire en Html5 avec un signe de fermeture.

Ainsi, la balise image <img> peut également s'écrire en Xhtml <img />.

L'espace avant la barre oblique de fermeture est important pour des raisons de compatibilité avec les (très) anciens navigateurs.

Ce qui nous handicape un peu, nous les francophones, c'est le fait que ces balises font appel à des termes ou des abréviations de termes anglo-saxons qui les rendent (au premier abord) abstraites et donc complexes.

*Exemples*

<b>	<b>b</b> pour <i>bold</i> ce qui signifie gras
<i>	<i>i</i> pour <i>italic</i> ce qui signifie italique
<p>	<b>p</b> pour <i>paragraph</i> ce qui signifie paragraphe
<div>	<b>div</b> pour <i>division</i> ce qui signifie ... division

<b>&lt;table&gt;</b>	<b><i>table</i> signifie tableau</b>
<b>&lt;form&gt;</b>	<b><i>form</i> signifie formulaire</b>
<b>&lt;img /&gt;</b>	<b><i>img</i> pour image</b>
<b>Etc.</b>	

## 2. Les attributs de la balise

**Il est parfois nécessaire de compléter une commande par des spécifications plus précises dans l'un ou l'autre domaine. Pour ce faire, le langage Html5 dispose des attributs de la balise. L'attribut s'insère dans la balise, entre le mot de commande et le signe > final.**

**La syntaxe complète d'une balise avec un attribut est : <balise attribut="valeur"> ... </balise>**

**L'attribut comporte toujours une valeur, celle-ci s'indique en complément de l'attribut par un signe égal (=) suivi de la valeur mise entre des guillemets. Il est impératif de mettre cette valeur entre guillemets. Le strict respect de la syntaxe veut qu'il n'y ait pas d'espace avant et après le signe égal.**

**Il est possible d'utiliser plusieurs attributs, séparés par un espace, dans une même balise : <balise attribut1="valeur" attribut2="valeur"> ... </balise>.**

## Le bon usage des balises

**Voici quelques règles simples qu'il faudra respecter lors de l'écriture des balises Html5.**

**En Html5, les balises ne sont pas sensibles à la casse. Ainsi, on peut écrire indifféremment <BALISE>, <Balise> ou <balise>. Certains voyaient même en l'utilisation des majuscules, un moyen efficace pour distinguer le code Html du contenu dans un document. Cependant, l'usage s'est généralisé d'écrire les balises en minuscules (comme en Xhtml).**

**La règle générale veut que toute balise ouverte <balise> doit être fermée </balise>. Les libertés prises dans l'écriture du Html 3.2 à cause du fonctionnement plus ou moins permissif des navigateurs ne sont plus d'actualité. La rigueur apportée par le Html 4.0 strict et le Xhtml 1.0 doit continuer à guider votre écriture.**

**Les balises doivent être correctement imbriquées. Lorsqu'on affecte plusieurs balises à un élément, l'ordre de fermeture de celles-ci est essentiel. La première balise de fermeture doit correspondre à la dernière balise d'ouverture non fermée. Un exemple et tout sera beaucoup plus clair :**

**Est correct :** <a><b><c>contenu</c></b></a>.

**Est incorrect :** <a><b><c>contenu</a></c></b>.

**Les valeurs des attributs doivent toujours figurer entre des guillemets. Ici aussi, la rigueur dans le code reste de mise.**

## Les balises et attributs Html 4.0 disparus

Les différences, tant au niveau des balises que des attributs, sont nombreuses et importantes par rapport au Html 4.0.

### 1. Au niveau des balises

Notons tout d'abord la disparition complète des cadres. Les balises `<frame>`, `<frameset>` et `<noframes>` ont définitivement disparu des outils à la disposition des auteurs. Ceci ne constitue pas réellement une surprise car ils étaient décriés depuis de nombreuses années. Les concepteurs professionnels les évitaient comme la peste depuis que Google avait annoncé à l'attention des webmestres qu'il ne pouvait garantir la bonne indexation des sites comportant des cadres. La balise `<iframe>` quant à elle subsiste en Html5.

Ensuite, la disparition de certaines balises de présentation comme `<big>`, `<center>`, `<font>`, `<strike>`, `<tt>` et `<u>`. Ici aussi, pas de grande surprise car le principe de la séparation de contenu et de la présentation est à présent bien établi. Ces balises de présentation doivent maintenant être prises en charge par des feuilles de style CSS.

### 2. Au niveau des attributs

Le Html5 va au bout du principe de la séparation du contenu et de la présentation. Ce qui étonne, c'est l'ampleur des attributs concernés. En résumé, ce sont tous les attributs relatifs à l'alignement, la largeur, les arrière-plans (de couleur ou avec image), les bordures et la numérotation des listes qui sont passés à la trappe.

Tous ces attributs doivent être pris en charge par les feuilles de style qui deviennent ainsi indissociables du code Html5.

De façon plus détaillée :

align dans les balises `<caption>`, `<iframe>`, `<img>`, `<input>`, `<object>`, `<legend>`, `<table>`, `<hr>`, `<div>`, `<hx>`, `<p>`, `<col>`, `<colgroup>`, `<tbody>`, `<td>`, `<tfoot>`, `<th>`, `<thead>` et `<tr>`.

alink, link, text et vlink dans la balise `<body>`.

background dans la balise `<body>`.

bgcolor dans les balises `<table>`, `<tr>`, `<td>`, `<th>` et `<body>`.

border dans les balises `<table>` et `<object>`.

**cellpadding et cellspacing dans la balise <table>.**

**frameborder dans la balise <iframe>.**

**height dans les balises <td> et <th>.**

**hspace et vspace dans les balises <img> et <object>.**

**marginheight et marginwidth dans la balise <iframe>.**

**noshade dans la balise <hr>.**

**nowrap dans les balises <td> et <th>.**

**rules dans la balise <table>.**

**size dans la balise <hr>.**

**type dans les balises <li>, <ol> et <ul>.**

**valign dans les balises <col>, <colgroup>, <tbody>, <td>, <tfoot>, <th>, <thead> et <tr>.**

**width dans les balises <hr>, <table>, <td>, <th>, <col>, <colgroup> et <pre>.**

**Ainsi, il n'est plus possible de fixer, en Html5, les bordures, l'alignement, les arrière-plans éventuels et la largeur des tableaux sans avoir recours aux feuilles de style CSS.**

**Avec la disparition des attributs color ou bgcolor, le Html5 se décline en noir et blanc.**

**Cette absence complète d'attributs de présentation nous a obligés de revoir notre structure d'apprentissage traditionnelle du Html avec d'un côté les balises Html et par ailleurs les feuilles de style CSS. Ainsi, dès les premiers pas en Html, nous avons adjoint des notions élémentaires de feuilles de style pour la présentation. Pour permettre à l'apprenant de s'exercer, par exemple, dans l'apprentissage des tableaux, des éléments de feuilles de styles se révélaient indispensables pour au moins visualiser ceux-ci par une bordure. Une étude plus poussée et plus détaillée des propriétés CSS est abordée comme d'habitude à la suite de l'étude des balises et attributs du Html5.**

## Les navigateurs de notre étude

**Ces technologies de pointe que sont le Html5 et les feuilles de style CSS3 ne sont reprises que par les dernières versions des navigateurs.**

**En outre, l'intégration du Html5 et des CSS3 n'est encore que partielle et varie d'un navigateur à l'autre. À ce stade, il est ainsi nécessaire de retenir pour notre étude les principaux acteurs de ces techniques innovantes pour assurer une vue d'ensemble des apports importants qu'ils entraînent ou entraîneront dans l'édition et la publication des documents Web.**

**Une intégration complète du Html5 et des CSS3 par les navigateurs de pointe prendra encore quelques mois, voire quelques années. Mais étant donné l'implication des firmes elles-mêmes dans l'élaboration du Html5, on peut présager que les évolutions seront rapides.**

**Il faudra également attendre un certain nombre d'années avant que, par le système de renouvellement ou de mises à jour des navigateurs, le Html5 et les feuilles de styles CSS3 deviennent des applications grand public.**

**Il est intéressant de noter l'importance donnée soudainement au JavaScript dans la conception des pages ou plutôt des applications Web. Cette prise de position se concrétise dans le fait que tous les navigateurs se sont dotés d'un nouveau moteur JavaScript. Que ce soit Opera 10.5 avec Carakan, Safari avec Nitro, Google Chrome avec V8 ou Internet Explorer 9 avec Chakra. On annonce même deux moteurs JavaScript dans Firefox 4 avec SpiderMonkey et le tout nouveau Narcissus.**

**Notons aussi l'apparition de l'accélération matérielle qui fait supporter par la carte graphique et non plus par le processeur, la restitution des animations et vidéos.**

### Internet Explorer 9

**Internet Explorer 8 est arrivé à bout de course et présente un retard conséquent par rapport à la concurrence en raison de sa prise en charge quasi nulle des nouveaux standards Html5 et des feuilles de style CSS3.**

**Internet Explorer 9, annoncé courant 2011, se repositionne dans le groupe des navigateurs modernes en assimilant (enfin) du Html5 et une partie des spécifications CSS3.**

**Un problème de taille subsiste toutefois puisqu'Internet Explorer 9 n'est prévu que sous Microsoft Vista et Seven. Donc pas d'Internet Explorer 9 pour les utilisateurs de XP. Cet abandon de XP s'explique du côté de Microsoft par l'accélération matérielle**

**d'Internet Explorer 9 qui réclame un système d'exploitation moderne. Pourtant, les spécialistes rétorquent que des navigateurs modernes comme Firefox, Chrome et Opera font bien de l'accélération matérielle sous XP. Cette prise de position est en tout cas regrettable pour les 50 % d'utilisateurs restés fidèles à XP. Mais c'est peut-être cette fidélité qui est reprochée...**

**Quelques mots pour terminer sur le cauchemar des développeurs Web. En effet, les statistiques indiquent que fin 2010, encore 5 % des internautes utilisent le totalement obsolète Internet Explorer 6.**

#### **Firefox 4**

**Attendu début 2011, Firefox 4 restera sans nul doute le navigateur préféré des concepteurs grâce à son système d'extensions qui en font le partenaire incontournable dans le développement d'applications Web. On pense spécialement à *Firebug* et *Web Developer* qui sont des extensions devenues quasi indispensables pour de nombreux développeurs. Les versions bêta disponibles à ce jour font apparaître de réelles avancées par rapport à son prédécesseur Firefox 3.6 que ce soit par son interface renouvelée, une meilleure prise en charge des standards HTML5, les CSS3, l'accélération graphique, etc.**

**Firefox 3.6, vieillissant par rapport à la concurrence et spécialement celle de Google Chrome, ne sera pas totalement oublié pour autant. Il faut garder à l'esprit qu'il reprenait déjà de nombreux effets des CSS3 alors que d'autres navigateurs (lire Internet Explorer) les ignoraient complètement.**

#### **Google Chrome 7**

**Apparu en septembre 2008, Google Chrome n'a cessé depuis d'accroître ses parts de marché. On peut considérer qu'il est à l'heure actuelle le troisième navigateur le plus utilisé sur la toile. Depuis les versions se sont succédées à un rythme soutenu et toujours à la pointe de technologies comme le HTML5 et la prise en charge progressive des feuilles de style CSS3. Il faut souligner que les versions récentes sont rapidement adoptées par les utilisateurs. Ce qui confirme la modernité et le dynamisme de ce jeune navigateur.**

#### **Safari 5**

**Safari est depuis 2003, le navigateur par défaut des plateformes Mac OS X. Depuis sa version 4 (juin 2009), il est également disponible pour Windows. Cet excellent navigateur, rapide et innovant, a cependant quelques difficultés à être adopté par les utilisateurs de Windows. Pourtant le navigateur Safari est à la pointe de la prise en charge du HTML5 et des feuilles de style CSS3 et ne concède rien en performances à la concurrence.**

#### **Opera 10.6**

**Ce navigateur sympathique mais plutôt confidentiel nous sera très utile pour notre étude du HTML5 car il est le plus novateur en la matière. Il nous permettra par exemple, d'illustrer les nouveautés**

**en matière de formulaires du Html5 qui ne sont pas encore reprises à ce jour par les navigateurs précités.**

# Le document Html5 minimum

## 1. Le doctype et son importance

**Tout document Html doit commencer par un doctype. Le Html5 propose un doctype unique et simplifié.**

```
<!DOCTYPE html>
```

**Pour se rendre compte de la simplification, il suffit de le comparer avec un doctype du Html 4.0.**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

**Nettement plus court et enfin mémorisable !**

**Le doctype, aussi appelé DTD sert à indiquer au navigateur à quelles règles d'écriture obéit le code source de la page Html ou Xhtml. Il utilisera celles-ci pour afficher la page selon les standards du W3C. Tous les navigateurs appliquant la même règle, vous êtes ainsi assuré d'un rendu identique entre les différents navigateurs.**

**En l'absence d'un doctype, le navigateur ignore selon quelles règles il doit traiter la page. Il se rabat ainsi sur des procédures qui lui sont propres pour afficher vaille que vaille votre page Html. Ce qui peut entraîner des différences de restitution sensibles entre les différents navigateurs. Ce mode bancal est appelé mode compatibilité ou *quirks mode*.**

**Vous pouvez aisément vérifier dans quel mode (quirks ou respect des standards) se trouve le navigateur.**

**Dans Firefox 3.6+, accédez au menu Outils - Informations sur la page - onglet Général - Mode de rendu.**

**Pour Internet Explorer 8+, encodez javascript:alert(document.compatMode) dans la barre d'adresse. Si une fenêtre d'alerte indique CSS1Compat, il s'agit du mode standard. Si la fenêtre d'alerte affiche BackCompat, il s'agit du mode *quirks*.**

**Le doctype doit se situer à la première ligne du fichier Html. S'il y a n'importe quoi d'autre, même un simple espace ou une ligne blanche, certains navigateurs considéreront que la page n'a pas de doctype et afficheront celle-ci en quirks mode.**

## 2. La balise Html

**La balise <html> indique au navigateur qu'il s'agit d'un document Html.**

**La balise <html> est l'élément le plus haut ou l'élément racine du document. Elle prend donc place juste après la déclaration de doctype.**

```
<!DOCTYPE html>  
<html lang="fr">  
...  
</html>
```

**La balise <Html> peut prendre comme attribut lang="fr" qui spécifie que le document est en langue française. Cette information est fort prisée par les moteurs de recherche comme Google ainsi que par les synthèses vocales utilisées par les personnes malvoyantes.**

**À l'adresse [www.w3schools.com/tags/ref\\_language\\_codes.asp](http://www.w3schools.com/tags/ref_language_codes.asp), vous pouvez trouver une liste complète des autres langues référencées.**

### **3. L'en-tête du document**

**La balise <head> ... </head> qui se positionne juste après le doctype et la balise <html> contient ce qu'on appelle l'en-tête du document.**

**Dans cet en-tête du document se situe toute une série d'informations relatives au document lui-même, indépendamment du contenu de celui-ci.**

**Ces informations peuvent être de natures très diverses :**

**Le titre du document. Soit la balise <title> ... <title> reprise au point suivant.**

**Des déclarations ou des appels de feuilles de style CSS.**

**Des fonctions ou des liens vers des fichiers JavaScript.**

**Des informations à l'intention des moteurs de recherche pour le référencement de la page.**

**Des informations à l'intention des navigateurs.**

**Des balises méta qui reprendront quant à elles l'encodage, la description de la page, les mots-clés associés à la page, le nom de l'auteur, des mentions de copyright, etc.**

**Dans un souci de simplification, le Html5 spécifie des valeurs par défaut pour l'attribut type des scripts, des styles et les éléments link. Sauf si vous avez expressément besoin d'une valeur différente que celles définies par défaut, vous pouvez omettre en Html5 cet attribut de type.**

**Le JavaScript est défini comme langage de script par défaut.**

**Ainsi, en Html 4.0, on devait écrire :**

```
<script type="text/javascript">
```

```
...
```

```
</script>
```

**Et**

```
<script type="text/javascript" src="fichier.js"></script>
```

**En Html5, on peut se contenter de :**

```
<script>
```

```
...
```

```
</script>
```

**Et**

```
<script src="fichier"></script>
```

**Les CSS sont définis comme le langage de feuilles de style par défaut.**

**Ainsi, en Html 4.0, on devait écrire :**

```
<style type="text/css">
```

```
...
```

```
</style>
```

**Et**

```
<link rel="stylesheet" type="text/css" href="fichier.css">
```

**En Html5, on peut se contenter de :**

```
<style>
```

```
...
```

```
</style>
```

**Et**

```
<link rel="stylesheet" href="fichier.css">
```

**Notre document minimal Html5 devient :**

```
<!DOCTYPE html>
```

```
<html lang="fr">
```

```
<head>
```

```
...
```

```
</head>
```

```
...
```

```
</html>
```

Mis à part le titre de la page (voir ci-après), aucun élément compris entre les balises <head> ... </head> n'apparaît dans le navigateur.

#### 4. Le titre du document

La balise <title>, incluse dans l'en-tête du document, est la seule balise obligatoire de celui-ci.

La balise <title> ... </title> donne un titre à votre page.

Ce titre est affiché dans la barre de titre, située en haut à gauche de tout navigateur et/ou dans les onglets ouverts par celui-ci.

La balise <title> a une place importante dans l'algorithme de référencement de Google. On veillera à avoir pour les documents Web, un titre accrocheur et synthétique. Il n'est pas inutile non plus d'y inclure un ou des mot(s)-clé(s) relatif(s) à la page.

Le document Html5 se complète ainsi :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Html5</title>
</head>
...
</html>
```

Si par mégarde il y avait une erreur dans les balises <title> ... <title>, une simple page blanche est affichée.

#### 5. L'encodage (*charset*) du document

La notion du jeu de caractères utilisé (*charset*) est une notion essentielle dans le développement de pages Web. Cette notion est pourtant parsemée d'embûches, surtout en termes d'interopérabilité. N'avez-vous jamais reçu d'emails où certains caractères étaient remplacés par d'autres signes cabalistiques comme des □t□ ou vÃ©rification ?

Le jeu de caractères désigne la façon dont les caractères d'un alphabet donné sont convertis en octets dans un fichier informatique (et vice-versa). Certaines méthodes d'encodage sont spécifiques à un environnement informatique dans une langue ou un alphabet donné, d'autres sont multiplateformes et multilingues. Exemple d'encodage : ASCII, windows-1252, ISO-

## 8859-1, ISO-8859-15, UTF-8, etc.

<b>ASCII</b>	<b>Jeu de caractère basique. Il entraînera une interopérabilité maximale mais il faudra encoder les caractères spéciaux par des entités du genre &amp;eacute; (pour l'€).</b>
<b>windows-1252</b>	<b>Jeu de caractères utilisé par Windows et de nombreuses applications Microsoft. C'est la porte ouverte à d'autres systèmes.</b>
<b>ISO-8859-1</b>	<b>L'ISO-8859-1 est le codage par défaut sur de nombreux protocoles réseau, ce qui est le gage d'une excellente interopérabilité.</b>
<b>ISO-8859-15</b>	<b>Version actualisée de l'ISO-8859-1 qui comporte le signe €.</b>
<b>UTF-8</b>	<b>Application de l'Unicode qui est présentée comme la solution pour l'avenir. Son adoption est à l'heure actuelle freinée en charge problématique dans certains langages de programmation (PHP).</b>

**Les navigateurs possèdent bien une fonction de détection automatique du jeu de caractères. Mais l'encodage ainsi retenu de façon arbitraire est parfois la porte ouverte à des erreurs d'interprétation.**

**Des surprises peuvent ainsi apparaître. Vous pouvez lire à ce sujet l'excellent article et la discussion qui s'en suit : "ISO-8859-1, ISO-8859-15, UTF-8, lequel choisir ?" à l'adresse <http://forum.alsacreations.com/topic.php?fid=17&tid=1201>.**

**Il fait ainsi partie des bonnes pratiques de la publication sur le web de définir prioritairement l'encodage du document. Celui-ci se réalise en Html5 par la balise :**

**<meta charset="iso-8859-1"> ou <meta charset="UTF-8">**

**L'écriture suivante, rappelant le Xhtml 1.0, est également acceptée.**

**<meta charset="iso-8859-1" /> ou <meta charset="UTF-8" />**

**Ainsi, il n'y a pas que le doctype qui est simplifié en Html5. Pour rappel, cette déclaration d'encodage prenait en Html 4.0 la forme :**

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

**Notre document Html5 minimal est alors :**

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Html5</title>
<meta charset="iso-8859-1">
</head>
...
</html>
```

**La spécification Html5 du W3C, suivant en cela l'IETF (Internet Engineering Task Force) qui édicte les standards Internet, recommande l'usage de l'utf-8. Rien n'empêche cependant de retenir le format iso-8859-1 plus fréquemment utilisé en Europe occidentale.**

## **6. Le corps du document**

**Les balises <body> ... </body> déterminent ce que l'on appelle le corps du document. C'est entre ces balises que prendra place le code Html5 qui sera utilisé pour élaborer le contenu de la page.**

**C'est cette partie du document Html qui sera affichée dans la fenêtre du navigateur et donc visible par l'internaute.**

**En Html5, les attributs dédiés à la présentation du corps du document de la balise <body> sont maintenant abandonnés pour laisser place à l'utilisation des feuilles de style.**

**Ces attributs qui déterminaient la couleur de l'arrière-plan (bgcolor), l'image d'arrière-plan (background) ou la couleur du texte (text) n'ont plus leur place en Html5.**

**Au final, notre document Html5 minimal se présente ainsi :**

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Html5</title>
<meta charset="iso-8859-1">
</head>
<body>
</body>
</html>
```

## La façon de procéder

**Les langages utilisés sur le Web ont cet avantage qu'ils ne réclament pas nécessairement un logiciel dédié. Pour écrire en code source, pas besoin de logiciel coûteux ! Un simple éditeur de texte, présent dans tous les systèmes d'exploitation, fait parfaitement l'affaire.**

**L'important est d'avoir du texte brut sans aucune mise en forme.**

**Sous Windows, vous pouvez utiliser le Bloc-notes ou Notepad. Pour rappel, on y accède par** Tous les programmes - Accessoires - Bloc-notes.

**Cet éditeur de texte aux fonctions rudimentaires est cependant "parfait" pour encoder du texte brut (sans aucun formatage).**

**Il est important de respecter une procédure d'enregistrement correcte. Après avoir fait** Fichier - Enregistrer sous..., **veillez à activer dans la zone** Type **l'option** Tous les fichiers.

**Dans le cadre de cet ouvrage consacré au Html5, les fichiers auront comme extension \*.htm ou \*.html. L'usage de l'extension \*.htm semble se généraliser.**

**Pour afficher votre fichier, il suffit d'ouvrir votre navigateur et d'indiquer le chemin vers le fichier concerné.**

**Cette opération semble toute simple. Cependant, avec la tendance d'une interface de plus en plus simplifiée des navigateurs, elle n'est pas toujours intuitive. Pour Firefox, Safari et Opera, il faut dans un premier temps activer la barre des menus, puis retrouver le fichier par** Fichier - Ouvrir un fichier. **Pour Internet Explorer 9 et Google Chrome dont l'interface est encore plus dépouillée, il n'y pas d'autre alternative que le raccourci clavier [Ctrl] O.**

**En cas de modifications apportées au code source initial, n'oubliez pas de réactualiser la page pour que le navigateur tienne compte de celles-ci.**

*Exemple*

**À ce stade, tous les outils sont réunis pour réaliser notre premier document Html5.**

```
<!DOCTYPE html>  
<html lang="fr">
```

```
<head>
<title>Html5</title>
<meta charset="iso-8859-1">
</head>
<body>
Un premier document Html5
</body>
</html>
```

**Ce qui s'affiche dans Firefox 4.0 :**

## La validation du code Html5

**Bien que le Html5 soit moins formaliste que le Html 4.0 strict ou le Xhtml 1.0, il reste utile voire indispensable de valider le code source.**

**Les avantages d'un code parfait et donc de la validation sont :**

**Une vérification en profondeur du Html5 pour les codeurs débutants (et confirmés).**

**L'assurance que votre page sera toujours correctement affichée par la plupart des navigateurs. Une page invalide peut amener les navigateurs à interpréter celle-ci de manière très différente.**

**La démonstration de vos capacités professionnelles en fournissant un code de qualité, conforme au standard du html5.**

**Les synthèses vocales et autres aides techniques à l'attention des personnes moins valides se basent sur un code valide pour restituer les pages de la toile.**

**Si le document soumis au validateur n'est pas correct, celui-ci vous fournira une analyse des problèmes qui vous permettra de corriger votre code. Les premières validations peuvent être très frustrantes mais constituent un excellent apprentissage du code source. Le rapport d'erreur est cependant souvent très technique.**

**Pour effectuer leur fonction de vérification et de validation, les validateurs ont besoin du doctype du document (voir section Le document Html5 minimum - L'encodage (charset) du document du présent chapitre).**

**Les validateurs Html 5 (en ligne) disponibles sont :**

**Le validateur du W3C lui-même.**

**Le validateur du W3Québec.**

**Le validateur validator.nu.**

**Le validateur du W3C**

**Le W3C lui-même présente un validateur de code (validator.w3c.org). Ce qui ne peut être qu'un gage de qualité.**

**La validation du Html5 est cependant encore annoncée comme**

**étant au stade expérimental.**

**Notons :**

**Qu'il est en anglais.**

**Qu'il offre de valider un fichier par son adresse http, un fichier présent sur votre ordinateur ou du code inséré dans une zone de texte.**

**Qu'il est très pointilleux.**

**Que ses remarques et explications sont très techniques voire parfois sibyllines.**

**Qu'il est "la" référence des professionnels.**

**Le validateur du W3Québec**

**Le W3Québec est un organisme sans but lucratif visant à promouvoir l'ensemble des normes, standards ouverts et bonnes pratiques du Web et du multimédia.**

**Il propose un validateur :**

**En français.**

**Avec la possibilité de valider un fichier par son adresse http, un fichier présent sur votre ordinateur ou du code inséré dans une zone de texte.**

**Les avertissements ou erreurs sont fournis en français dans un langage technique mais accessible.**

**Il manque parfois des explications plus explicites pour corriger le code soumis.**

**Il s'agit en fait d'une traduction en français du validateur du W3C.**

**Il faut noter que le validateur du W3Québec ([www.w3qc.org/validateur/](http://www.w3qc.org/validateur/)) était en reconstruction lors de la conception de cet ouvrage.**

**Le validateur [validator.nu](http://validator.nu)**

**Le site [validator.nu](http://validator.nu) propose également au stade expérimental un validateur Html5 ([html5.validator.nu/](http://html5.validator.nu/)).**

**Signalons qu'il fut le premier à proposer un validateur Html5.**

**On peut souligner :**

**Qu'il est en anglais.**

**Que son interface est très simplifiée.**

Qu'il offre de valider un fichier par son adresse http, un fichier présent sur votre ordinateur ou du code inséré dans une zone de texte.

Qu'il semble d'excellente qualité.

Pour les amateurs d'extensions Firefox, le module *Html5 Validator* vous permet, après installation, de valider votre page via le menu contextuel par le validator.nu.

### Test

Nous n'avons pas résisté à la tentation de réaliser un test à partir d'un code html5 erroné.

Le code erroné est :

```
1. <!DOCTYPE html>
2. <html lang="fr">
3. <head>
4. <title>Html5</title>
5. <meta charset="iso-8859-1">
6. </head>
7. <body>
8. <p align="center">
9. <s>Html5</s>
10. </p>
11. <table width="200px">
12. <tr>
13. <td>1</td><td>2</td>
14. </tr>
15. <tr>
16. <td>3</td><td>4</td>
17. </tr>
18. </table>
19. </body>
20. </html>
```

Les erreurs sont :

À la ligne 8, l'usage de l'attribut align qui est supprimé (obsolète) en Html5.

À la ligne 9, l'usage de la balise de soulignement <s> qui n'existe plus en Html5.

À la ligne 11, l'usage de l'attribut width de la balise <table> est supprimé en Html5.

Le validateur du W3C détecte sans difficultés ces erreurs.

Line 8, Column 18: <p align="center">	The align attribute on the p element is obsolete. Use CSS instead.
Line 9, Column 3:	The s element is obsolete. Use CSS

<code>&lt;s&gt;Html5&lt;/s&gt;</code>	<b>instead.</b>
<b>Line 11, Column 21:</b> <code>&lt;table width="200px"&gt;</code>	<b>The width attribute on the table is obsolete. Use CSS instead.</b>

**L'affichage de ce code erroné nous fournit une belle justification de l'utilité d'une validation du code. En effet, tous les navigateurs de notre étude affichent le terme Html5 barré alors qu'un soulignement était prévu par la balise `<s>`.**