

Les standards du web

par Russ Weakley ([Max Design](#)) Normand Lamoureux (traducteur) ([Normand Lamoureux](#))

Date de publication : 26/03/2008


Dernière mise à jour : 3/12/2008

Les avantages des standards web pour vos visiteurs, vos clients et vous !

I - Les standards web.....	3
II - Qu'est-ce que les standards web ?.....	3
III - Un changement de mentalité.....	3
III-A - Site web traditionnel.....	3
III-B - Site web conforme aux standards.....	4
IV - Un balisage sémantiquement correct.....	4
V - Qu'est-ce qu'un code valide ?.....	4
VI - Pourquoi utiliser un code accessible ?.....	5
VII - Pourquoi utiliser des CSS pour séparer la présentation du contenu ?.....	5
VIII - La puissance des CSS en action.....	7
IX - Comment vos visiteurs profitent-ils des standards web ?.....	7
X - Comment vos clients profitent-ils des standards web ?.....	7
XI - Comment profitez-vous vous-mêmes des standards web ?.....	8
XII - Quels sont les inconvénients ?.....	8
XIII - Comment appliquer les standards web ?.....	8
XIV - Conclusion.....	9
XV - Ressources sur les standards web.....	9

I - Les standards web

« Les standards web veulent servir de **base commune**... de fondement au web pour que les navigateurs et les autres logiciels puissent comprendre **le même vocabulaire de base**. » *Eric Meyer*

Le  **W3C** (World Wide Web Consortium) et d'autres organisations ont défini des **technologies** pour créer et interpréter les contenus web. Les standards actuels sont :

Langages structuraux :

- [Le langage de balisage hypertexte extensible \(XHTML\) 1.0 \[en\]](#)
- [XHTML 1.1 \[en\]](#)
- [Le langage de balisage extensible \(XML\) 1.0 \[en\]](#)

Langages de présentation :

- [Les feuilles de style en cascade \(CSS\) Niveau 1 \[en\]](#)
- [CSS Niveau 2](#)
- [CSS Niveau 3 \(document de travail\) \[en\]](#)

Modèle d'objets de document :

- [Le modèle d'objets de document \(DOM\) Niveau 1\[en\]](#)
- [DOM Niveau 2\[en\]](#)

Langages de script :

- [ECMAScript 262 \[en\]](#) (la version standard de JavaScript)

Langages de balisage complémentaires :

- [Le langage de balisage pour les mathématiques \(MathML\) 1.01 \[en\]](#)
- [MathML 2.0 \[en\]](#)
- [Les graphismes vectoriels redimensionnables \(SVG\) 1.0 \[en\]](#)

II - Qu'est-ce que les standards web ?

Les « standards web » sont conçus pour :

- procurer **le plus d'avantages** au plus grand nombre d'utilisateurs possible ;
- assurer **la viabilité à long terme** de n'importe quel document web ;
- **simplifier le code** et réduire les coûts de production ;
- construire des sites **accessibles** à un plus grand nombre de **personnes** et à une plus grande variété de **périphériques** Internet ;
- continuer de **fonctionner correctement** au fur et à mesure que les **navigateurs évoluent** et que de nouveaux périphériques font leur apparition.

Pour les designers et les développeurs web, les standards web consistent à **utiliser les standards existants** (les langages structuraux, de présentation, d'objets de document et de script) et à **adopter de meilleures pratiques** (écrire un code valide, sémantiquement correct et accessible)... pour la plus grande joie de tous.

III - Un changement de mentalité

III-A - Site web traditionnel

Le développement d'un site web traditionnel se situe dans le prolongement des **médias écrits** : il doit être conçu pour s'afficher **au pixel près** dans les 5 ou 6 principaux navigateurs.

Quelques caractéristiques communes à ce genre de site :

- mises en page construites sur des **tableaux** ;
- instructions de présentation à **même** le contenu (balises) ;
- code **invalide** ;
- code **inaccessible** ;
- code sémantiquement **incorrect**.


III-B - Site web conforme aux standards

Les standards web supposent de voir le web comme un moyen de communication puissant pouvant être utilisé par une grande variété d'**utilisateurs** et un large éventail de **périphériques**.

Quelques caractéristiques communes à ce genre de site :

- **balisage sémantiquement correct** ;
- **code valide** ;
- **code accessible (aux personnes et aux périphériques)** ;
- **présentation séparée du contenu grâce aux feuilles de style en cascade (CSS)**.

IV - Un balisage sémantiquement correct

Le balisage sémantiquement correct fait usage des **éléments HTML** conformément à leurs **finalités propres**. Le  **HTML** bien structuré possède une signification sémantique pour un large éventail d'agents utilisateurs (navigateurs sans feuilles de style, navigateurs textuels, PDA, engins de recherche, etc.).

Pour le balisage, il faut utiliser les éléments HTML standards et éviter de styler d'autres éléments HTML pour **les faire ressembler aux premiers**. En termes simples, cela signifie :

- pour les en-têtes, utiliser les éléments d'en-tête en commençant par <h1> ;
- pour les paragraphes de texte, utiliser les éléments de paragraphe ;
- pour les listes, utiliser les éléments de liste.

V - Qu'est-ce qu'un code valide ?

La validation est un processus qui consiste à **mesurer la conformité d'un document** à l'aune d'une **norme officielle** semblable à celles qui ont été émises par le W3C. Un document qui a été vérifié et qui a passé le test est considéré comme valide.

Pourquoi utiliser un code valide ?

- parce qu'un code valide s'affichera **plus rapidement** qu'un code avec des erreurs ;
- parce qu'un code valide s'affichera **mieux** qu'un code invalide ;
- parce que plus les navigateurs deviennent conformes aux standards, plus il est **impératif** d'écrire un HTML conforme et valide.

Comment vérifier validité de votre code ?

- **le service de validation HTML du W3C** [[en](#)] ;
- **le service de validation CSS du W3C** [[en](#)] ;
- **le validateur HTML du WDG** [[en](#)] ;
- **le validateur CSS de Stylesheets.com** [[en](#)] ;
- plusieurs logiciels de production sont maintenant dotés de leur propres outils de validation.

Comment faire un code valide ?

- commencer par un **doctype** approprié ;

- prendre garde aux différents **modes de fonctionnement** des doctypes (conforme aux standards, capricieux, etc.) ;
- utiliser une table de caractères définie ;
- fermer tous les éléments HTML ;
- ajouter l'attribut "alt", même s'il n'a pas de valeur, à chaque image ;
- faire régulièrement usage d'un validateur HTML ;
- régler toutes les irrégularités décelées avant de publier ;
- introduire la validation dans sa méthodologie de travail.

VI - Pourquoi utiliser un code accessible ?

- parce que cela rend votre site accessible à une **plus large auditoire** (handicapés visuels, handicapés moteurs, handicapés cognitifs, etc.) ;
- parce que cela rend votre site accessible à un plus large éventail de **périphériques** (portables, lecteurs d'écrans, navigateurs textuels, engins de recherche, etc.) ;
- parce que c'est **exigé par la loi** pour tout site gouvernemental (notamment aux États-Unis et au Royaume-Uni).


Comment faire un code accessible ?

- associer un **équivalent textuel** à tout contenu non-textuel ;
- construire des **tableaux de données accessibles** (en identifiant les en-têtes de lignes et de colonnes, etc.) ;
- élaborer des **formulaires accessibles** (en utilisant "label for", "id", <fieldset>, <legend>, etc.) ;
- utiliser du **texte balisé** plutôt que des images pour transmettre de l'information ;
- fournir un lien permettant de **sauter le menu** ;
- définir des **clés d'accès** ;
- utiliser des **unités de mesure relatives** pour contrôler la mise en page et la présentation ;
- s'assurer que les documents peuvent être consultés **indépendamment des feuilles de style** ;
- définir des **métadonnées** pour ajouter des informations d'ordre sémantique.

VII - Pourquoi utiliser des CSS pour séparer la présentation du contenu ?


Le développeur web doit chercher à soustraire toute instruction de présentation du code HTML, écrire un code propre et sémantiquement correct.

Avantages :

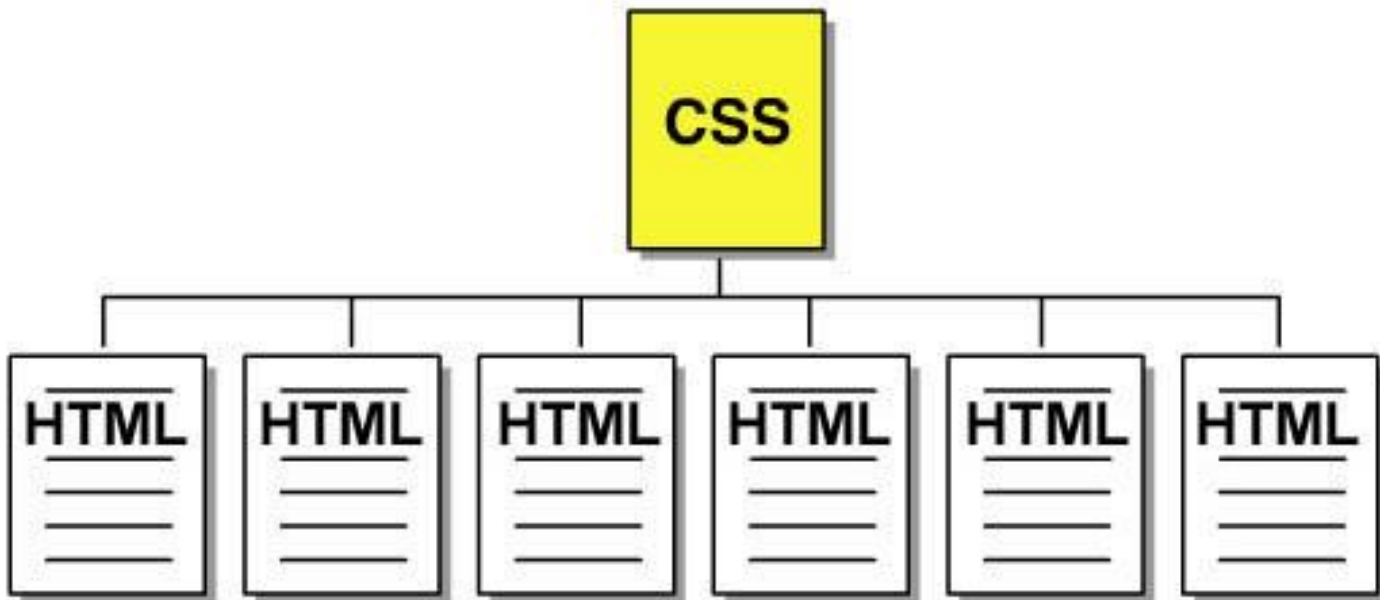
- **accessibilité du code** à une plus grande variété de périphériques ;
- plus facile de faire des **changements d'ensemble** (un seul fichier  **CSS** à modifier plutôt que toutes les pages une à une) ;
- moins de code sur la page, donc de **plus petits fichiers** et un **chargement plus rapide** ;
- **moins de code** sur la page, donc plus facile de coder ;
- permet aux usagers de **choisir un design** en fonction de leurs besoins (changeur de style) ;
- **contrôle accru** sur le code (possibilité d'ordonner le code pour les lecteurs d'écran).

Présence d'instructions de présentation sur toutes les pages HTML

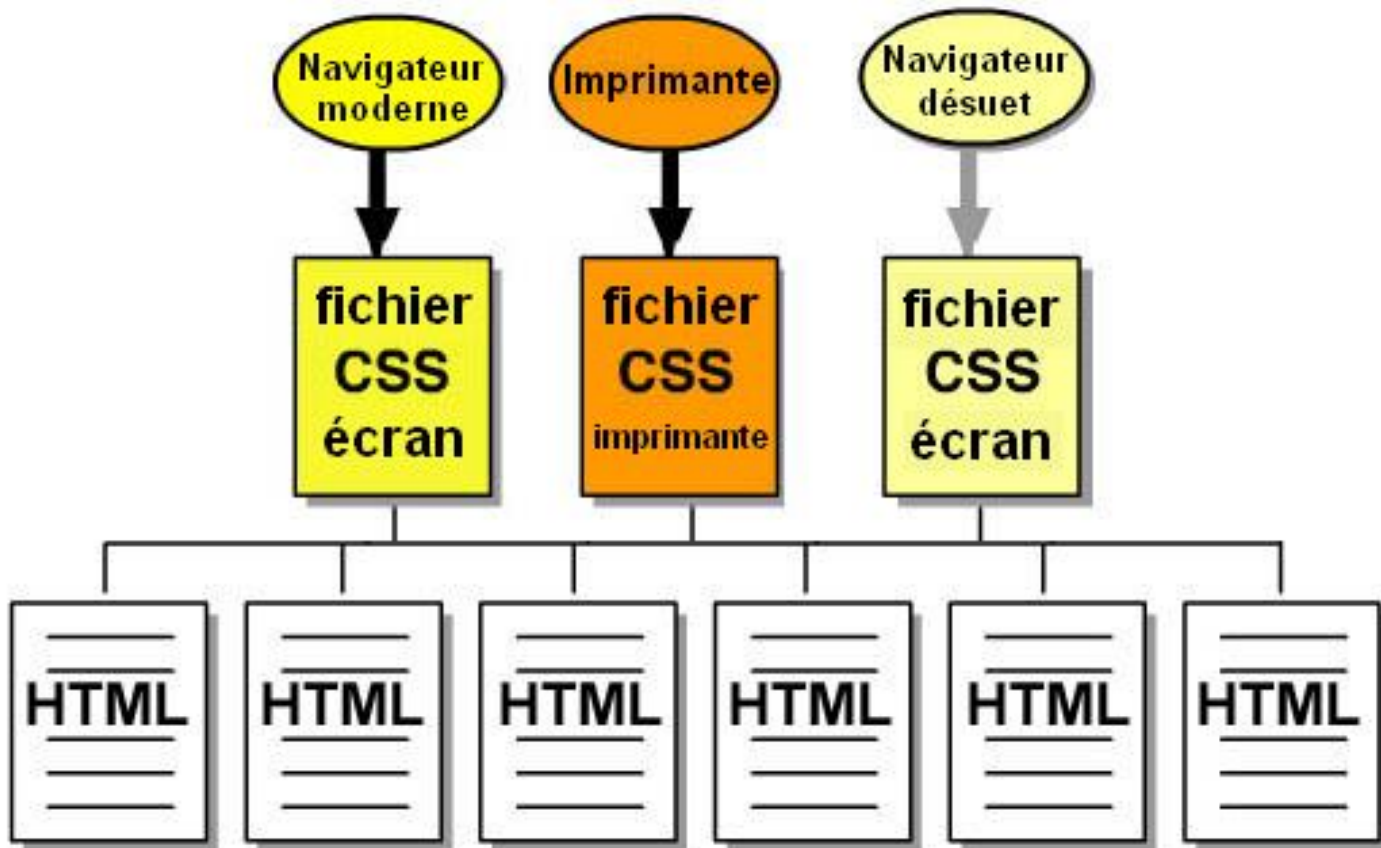


 = Instructions de présentation

Les instructions de présentation sont groupées dans un fichier CSS séparé



Plusieurs fichiers CSS pour contrôler séparément les navigateurs modernes, les imprimantes et les navigateurs désuets



VIII - La puissance des CSS en action

Un des avantages des CSS réside dans le fait de pouvoir **transformer l'apparence de tout un site** sans toucher à une seule ligne de code HTML.

Exemple d'un même contenu :

- **sans feuilles de style ;**
- **sans feuilles de style, mais avec des conteneurs <div> ;**
- **en HTML ;**
- **habillé d'un design en CSS ;**
- **habillé d'un autre design en CSS ;**
- **en sa version imprimable**

IX - Comment vos visiteurs profitent-ils des standards web ?

- **chargement plus rapide** des fichiers (moins de code, pas de tableaux imbriqués, code valide, etc.) ;
- contenus accessibles à une plus grande variété **d'utilisateurs** (gens normaux, aveugles, déficients visuels, dyslexiques, handicapés moteur, etc.) ;
- contenus accessibles à un plus large éventail de **périphériques** (lecteurs d'écran, navigateurs, navigateurs textuels, portables, engins de recherche, imprimantes, réfrigérateurs, etc.) ;
- permet aux usagers de **personnaliser** l'apparence du site (changeur de style) ;
- offre une **version imprimable** pour chaque page.

X - Comment vos clients profitent-ils des standards web ?

- **facilité de maintenance** (moins de code, plus de lisibilité) ;

- **coûts d'hébergement réduits** (moins de code) ;
- meilleur **positionnement** dans les engins de recherche ;
- **design facilement modifiable** sans toucher au code ;
- offre des possibilités de **personnalisation** aux usagers ;
- offre une **version imprimable** sans avoir à dupliquer le contenu ;
- augmente **l'accessibilité** (nécessaire aux clients gouvernementaux).

XI - Comment profitez-vous vous-mêmes des standards web ?

- **codage** plus facile (modulaire) ;
- **maintenance** plus facile (moins de code, plus de lisibilité) ;
- moins de **dépendance aux technologies particulières** (stabilité du code à travers un plus large éventail de périphériques) ;
- **gain de compétitivité** dans un monde qui se tourne de plus en plus vers les standards web.

XII - Quels sont les inconvénients ?

- nécessite de **nouveaux apprentissages** ;
- ne surmonte pas les problèmes **d'incompatibilité** entre navigateurs ;
- certaines mises en page demeurent **plus faciles** à réaliser avec des tableaux qu'avec des CSS.

XIII - Comment appliquer les standards web ?

Les standards web ne sont pas à **prendre ou à laisser**. Tous les développeurs devraient tendre vers une **intégration progressive** des standards.

L'acquisition de nouvelles habitudes de travail prend **du temps** et demande **de la pratique**. Plutôt que de se précipiter et devenir vite frustré, mieux vaut **se fixer des buts réalisables** et passer aux standards web **graduellement**. Par exemple :

Changements de base

- mettre un doctype approprié à toutes les pages ;
- associer un texte "alt" à toutes les images ;
- rédiger des titres de page significatifs.

Changements intermédiaires

- valider le code ;
- coder d'une manière sémantiquement correcte ;
- remplacer les balises et les déclarations de couleur par des instructions CSS ;
- rendre les formulaires et les tableaux de données accessibles, et inclure un lien pour sauter le menu.

Changements avancés

- remplacer les mises en page simples par un positionnement CSS de base (padding, margin, etc.) ;
- mettre fin aux mises en page par tableaux et tout faire en CSS.

S'exercer à la mise en page CSS

- lire des tutoriels et des livres pour apprendre le langage CSS ;
- se réserver du temps pour apprendre le positionnement CSS à l'abri de toute pression ;
- essayer plusieurs types de mises en page CSS pour acquérir de la confiance en vous-mêmes.

XIV - Conclusion

Ce que les standards web promettent :

- un chargement plus rapide des fichiers ;
- un code accessible à un plus large éventail d'utilisateurs et de périphériques ;
- des possibilités de personnalisation par l'utilisateur ;
- l'implémentation facile de versions imprimables ;
- une diminution des coûts d'hébergement ;
- un meilleur positionnement dans les moteurs de recherche ;
- une maintenance plus rapide et plus efficace ;
- une longueur d'avance sur la concurrence.

Bref, les standards web avantagent vos visiteurs, vos clients et vous.

XV - Ressources sur les standards web

Les standards web

- <http://www.zeldman.com> [en]
- <http://webstandards.org> [en]
- <http://webstandardsgroup.org> [en]
- <http://www.nypl.org/styleguide/> [en]

Balisage sémantique

- <http://brainstormsandraves.com/articles/semantics/structure/> [en]

Validation

- <http://validator.w3.org/>[en]
- <http://www.htmlhelp.com/tools/validator/>[en]
- <http://webboy.net/presentation/ict2004/validation.htm>[en]
- <http://webboy.net/presentation/validation.cfm>

Accessibilité

- <http://www.joelclark.org/>[en]
- <http://www.accessify.com/>[en]
- <http://www.juicystudio.com/>[en]

CSS

- <http://webboy.net/presentation/ict2004/01.htm> [en]
- <http://webboy.net/presentation/ict2004/02.htm> [en]
- <http://webboy.net/presentation/ict2004/03.htm> [en]
- http://westciv.com/style_master/academy/css_tutorial/ [en]
- <http://css.maxdesign.com.au> [en]
- <http://www.thenoodleincident.com/tutorials/css/> [en]