



# Le W3C et les standards du web

---

## Informations sur le tutoriel

[Ajouter à mes tutoriels favoris](#)



Auteur : [M@teo21](#)

Licence :

[Plus d'informations](#)

## Popularité

Visualisations : 8 567 191

Appréciation 15  
des lecteurs :15  
60

888

## Publicité

## Historique des mises à jour

- *Le 08/08/2010 à 03:23:12*  
Correction conjugaison.
- *Le 03/08/2010 à 01:07:48*  
Correction orthographique, #2605
- *Le 31/07/2010 à 18:20:55*  
Correction orthographique, ticket n°2593

Voilà une annexe au nom bien pompeux, et pourtant terriblement importante.

Je préfère vous prévenir de suite : vous n'allez pas apprendre ici à créer des supers effets dynamiques qui clignotent dans tous les sens, mais vous allez apprendre à connaître un peu de l'histoire du web.

Pourquoi aujourd'hui on fait du XHTML et non plus du HTML ? Pourquoi on a décidé de séparer le contenu (XHTML) de la mise en forme (CSS) ?

Pourquoi on dit que certains sites sont valides alors que d'autres ne le sont pas ?

Et le votre de site... est-il valide ?

Sommaire du chapitre :



- [L'histoire du web](#)
- [Du HTML au XHTML](#)
- [Votre site est-il valide ?](#)

---

## L'histoire du web

Nous allons commencer par la base à connaître : l'histoire du web. Tout webmaster se doit de savoir depuis quand le web existe et quelle est sa petite histoire.

Ce n'est pas seulement de la "culture générale", c'est aussi ça qui va vous permettre de comprendre la suite de cette annexe. Soyez attentifs, et vous paraîtrez un peu moins bête quand vous discuterez du web avec vos amis webmasters 🤖

- Tout a commencé au début des années 70, au cours d'une réunion à l'imprimerie du gouvernement canadien. Un certain William Tunnicliffe propose de séparer le contenu de l'information (le texte) de la manière de l'afficher. Déjà à l'époque, alors qu'Internet n'était réservé qu'à quelques très rares privilégiés, on parlait de séparer le contenu (ce qui est aujourd'hui devenu le langage XHTML) de la mise en forme (qui est devenu le CSS).
- Les choses s'accélérent en 1978, des personnes se réunissent pour créer un "langage informatique standard" riche et évolutif.
- En 1986, le **SGML** finit par naître. C'est l'abréviation de *Standard Generalized Markup Language*. Effectivement, c'est un langage puissant, mais trop complexe. Pour tout vous dire, sa documentation comportait 1290 pages ! Pas de quoi motiver le grand public à son utilisation... Le SGML ne connaîtra donc jamais de grande gloire 😞
- En 1991, un certain Tim Berners-Lee invente le World Wide Web (abrégez WWW). Pour créer les premières pages web, Tim Berners-Lee crée le **langage HTML** qui s'inspire un peu du SGML (comme quoi il aura servi à quelque chose celui-là 😊)  
Le HTML (abréviation de *HyperText Markup Language*) est diffusé gratuitement (merci Tim !).



Tim Berners-Lee

- A peine 2 ans plus tard, un des premiers navigateurs gratuits apparaît : il s'appelle Mozaïc et fonctionne aussi bien sur Mac que sur PC.  
Entre 1993 et 1994, le nombre de sites web passe de 500 à 10 000, la croissance est énorme. Déjà, le Web semble être promis à un grand avenir.
- C'est en Europe que les choses commencent à se précipiter. Le Web connaît un essor fulgurant, et il n'y a pas vraiment d'organisme chargé de surveiller son évolution. Il faut vérifier que ça ne parte pas dans tous les sens, il faut que les gens travaillent ensemble pour assurer l'évolution du Web (et du HTML !)
- Tim Berners-Lee, celui qui a inventé le Web, crée en 1994 le **W3C** (abréviation de *World Wide Web Consortium*). Il en devient le directeur.  
Le W3C est un organisme dont la mission va être de "surveiller" l'évolution du web : il faut éviter que certaines grosses entreprises tentent de stopper son évolution pour des raisons commerciales. Le W3C aura aussi la tâche de proposer au fur et à mesure de nouvelles versions du HTML, justement afin que le Web évolue.



Le logo du W3C

- Il est à noter que, dès le début (1997), le nombre de français travaillant au W3C est important. Mais, petit à petit des universités un peu partout dans le monde vont contribuer au W3C, notamment les japonais.
- En 2003, ce sont plusieurs organismes qui travaillent pour le W3C. Il y a environ 450 membres.  
L'ERICM (*European Research Consortium for Informatics and Mathematics*) remplace l'INRIA (un institut français) dans la direction du W3C. L'ERICM est situé en France lui aussi, à Sophia Antipolis (près de Nice) et va permettre de donner plus de poids au W3C, encore aujourd'hui parfois royalement ignoré par les Webmasters (dont de nombreux sont des "professionnels").

Comme vous pouvez le voir, tout au long de cette petite histoire des gens ont fait attention à ce que le web se

développe correctement et qu'il ne soit pas le joujou de grosses entreprises multinationales. Aujourd'hui encore (et plus que jamais), le W3C a la mission d'assurer un avenir au Web. Nous allons justement voir quel a été son travail sur le langage HTML, et pourquoi *on parle aujourd'hui de standards avec le XHTML et le CSS*.

---

## Du HTML au XHTML

Comme je vous l'ai dit dans ma petite histoire, dès les débuts du Web Tim Berners-Lee crée la première version du langage HTML.

Car en effet, comme pour un programme informatique, le langage HTML a connu plusieurs versions :

- **HTML 1.0** : c'est la toute première version créée par Tim Berners-Lee en 1991.
- **HTML 2.0** : la deuxième version du HTML apparaît en 1994 et se finira en 1996 avec l'apparition du HTML 3.0. C'est cette version qui posera en fait les bases des prochaines version du HTML. Les règles et le fonctionnement de cette version sont données par le W3C (tandis que la première version a été créée par un seul homme).
- **HTML 3.0** : apparue en 1996, cette nouvelle version du HTML rajoute de nombreuses possibilités au langage comme les tableaux, les applets, les scripts, le positionnement du texte autour des images etc...
- **HTML 4.0** : c'est une des dernières versions du HTML (la toute dernière étant une version légèrement modifiée, le HTML 4.01). Elle apparaît pour la première fois en 1998, et propose l'utilisation de frames (qui découpent une page web en plusieurs parties), des tableaux plus complexes, des améliorations sur les formulaires etc...Mais surtout, cette version permet pour la première fois l'utilisation de feuilles de style, notre fameux CSS !
- **XHTML 1.0** : début 2000, le W3C décide de mettre un terme au joyeux bordel qu'est devenu le langage HTML. En effet, au fur et à mesure de son évolution, des balises HTML ont été "inventées" par les navigateurs Netscape et Internet Explorer (de Microsoft). Certaines balises marchaient sur un navigateur, mais pas sur l'autre ><

On décide d'arrêter le développement du langage HTML et d'en créer un nouveau. Le XHTML (*Extensible HyperText Markup Language*) devient alors le **standard** : ce langage doit fonctionner de la même manière sur tous les navigateurs, et pas le droit de créer de nouvelles balises tant que le W3C ne l'a pas autorisé ! Concrètement, par rapport au HTML il y a assez peu de différences, mais le langage est plus "Strict" : vous n'avez pas autant le droit à l'erreur qu'en HTML. Ca ne veut pas dire pour autant qu'il est plus difficile à utiliser, c'est juste une habitude à prendre.

Dans le cours de création de site web du Site du Zéro que vous êtes en train de lire, je ne parle que du XHTML car c'est le langage qu'il faut aujourd'hui utiliser. Je vous ai donc enseigné les bonnes habitudes dès le début, ce qui va vous faciliter grandement la création de vos sites web. Ceux qui ont commencé par apprendre le HTML ont en effet parfois quelques mauvaises vieilles habitudes.

C'est désormais l'avenir du Web, et il va permettre de créer des sites web consultables sur toutes sortes de PDA, téléphones multimédia et autres produits transportables qui vont se multiplier dans l'avenir.

Vous imaginez s'il y avait 35 formats de CD Audio différents ? Votre lecteur de CD pourrait en lire certains, mais pas tous... Je vous dis pas le bordel que ça serait. Eh bien c'est pareil pour le XHTML : des gens se sont réunis et se sont mis d'accord pour dire comment doit fonctionner une page web. Au final, tout le monde y gagne : vous (le webmaster) et eux (vos visiteurs).

Créer un standard, ça permet donc de s'assurer que tout le monde parle et comprend le même langage. C'est toujours en inventant des standards (et pas seulement en informatique) qu'on arrive à faire évoluer les choses ensemble 😊

## Et le CSS ?

L'histoire du CSS débute, elle, en 1996. On revient à l'idée lancée par le monsieur dont je vous ai parlé au tout début de cette annexe (William Tunnicliffe) : il faut séparer le contenu de la mise en page. Cela apporte de nombreux avantages : l'apparence du site web pourra être plus facilement mise à jour, les pages seront plus rapides à charger, on pourra proposer plusieurs designs aisément etc...

- **CSS 1** : dès 1996, la première version du CSS est utilisable. Elle pose les bases de ce langage qui permet de présenter sa page web, comme les couleurs, les marges, les polices de caractères etc...

- **CSS 2** : apparue en 1999, cette nouvelle version de CSS rajoute de nombreuses options très intéressantes. On peut désormais utiliser le positionnement absolu, les pseudo-formats `:before` et `:after` etc etc...

---

## Votre site est-il valide ?

Pour la dernière partie de ce chapitre, on arrête un peu le blabla et on retourne à la technique.

Ce que je vous ai dit jusqu'ici avait pour but de vous faire connaître un peu mieux l'histoire du web, mais aussi de vous sensibiliser à ce qui va suivre...

Qu'est-ce que c'est que cette histoire de validité ?

Il y a des sites qui sont valides et d'autres qui ne le sont pas ?

Souvenez-vous : le W3C a établi des normes. Il est nécessaire de les respecter, pour qu'on soit sûrs que tous les sites web parlent la même "langue".

Le W3C propose sur son site web ([www.w3.org](http://www.w3.org)) un outil qui s'appelle le "Valideur" ("Validator" en anglais :p). Le valideur est une sorte de programme qui va analyser votre code source et vous dire s'il est correctement fait, ou s'il comporte des erreurs que vous devez corriger.

Il existe 2 valideurs différents :

- Un valideur (x)HTML.
- Un valideur CSS.

Nous allons apprendre à nous servir des 2 en commençant par le valideur XHTML.

Pour information, le valideur XHTML risque de vous montrer plusieurs erreurs sur votre page web. Ne vous inquiétez pas, c'est normal la première fois 😊

En revanche, le valideur CSS ne nous posera que peu de problèmes, il est plus rare de faire des erreurs en CSS qu'en XHTML.

### Le valideur XHTML

Commençons par le gros morceau voulez-vous 😊

Tout d'abord, commencez par mettre cette page en favoris, c'est l'adresse du Valideur XHTML :

<http://validator.w3.org/>.

Ceux qui ont téléchargé le plugin "Web Developer" pour Firefox peuvent d'ailleurs effectuer une validation automatiquement via un menu spécial.

La page que vous avez sous les yeux est très simple. Elle vous propose 2 façons de valider :

- Soit vous validez une page XHTML qui se trouve déjà sur le Web. Dans ce cas, il suffit d'indiquer l'URL de la page (par exemple : "http://www.monsite.com/page.html").
- Soit vous n'avez pas encore envoyé vos pages sur le Web (elles se trouvent donc toujours sur votre disque dur). Vous avez la possibilité d'envoyer directement la page au Valideur depuis votre disque pour qu'il l'analyse.

Dans les 2 cas, le résultat sera le même, il n'y a aucune différence.

Si tout se passe bien (vous avez beaucoup de chance 😊), vous verrez le message suivant :

**This Page Is Valid XHTML 1.0 Strict!**



Le site vous proposera alors de mettre l'icône ci-contre sur votre site web pour montrer que vos pages sont conformes aux normes, et donc que vous respectez les règles établies.

Ce n'est pas du tout une obligation, si vous trouvez ce logo trop moche rien ne vous impose de le mettre 😊

Bien entendu, si votre site web comporte plus d'une page (ce qui est très très fréquent ^^), il faudra tester chacune d'elles.

Au SEEECOUUUUUUURS !!! 🤔

Ma page web n'est pas valide, je vais pas m'en sortir je suis cerné par les erreurs, faites quelque chose aidez mmmoi!!!! 🤔

*C'était la réaction classique du débutant qui découvre que sa page web qu'il croyait parfaite ne l'est finalement pas tellement*

Tout d'abord, une chose à bien vous mettre dans la tête : *ce n'est pas parce que votre page web s'affiche correctement qu'elle ne comporte pas d'erreurs.*

Votre page web peut être toute belle et comporter pas mal d'erreurs.

Quel intérêt de les corriger alors ?

Je n'ai pas arrêté de vous le rappeler : si vous dites que votre page web parle le "XHTML 1.0", il faut qu'elle le parle correctement pour que tout le monde comprenne. Il faut savoir que les navigateurs "essaient" de ne pas afficher les erreurs lorsqu'ils en rencontrent pour ne pas perturber l'internaute, mais rien ne vous dit que d'autres navigateurs ne vont pas se comporter bizarrement !

Avoir une page web valide, c'est donc avoir la possibilité de dormir tranquille en sachant que l'on a bien fait les choses comme il faut.

De plus, et c'est vérifié, une page web correctement construite aura plus de chances d'être mieux positionnée dans les résultats de recherche de Google, ce qui vous amènera... plus de visiteurs !

Je ne vais pas vous lister toutes les erreurs possibles, ce serait trop long. En revanche, je vais vous faire une liste des erreurs les plus courantes. Vous vous apercevrez certainement que vous en avez commises quelques-unes, et vous saurez comment les corriger.

- Tout d'abord, ça peut paraître évident parce que je l'ai dit au début, mais il faut le rappeler : les noms de vos balises et attributs doivent être en **minuscules**. La valeur d'un attribut, en revanche, peut comporter des majuscules :

```
<balise attribut="Valeur de l'attribut">
```

- Les balises seules, comme le retour à la ligne "br" et la balise d'image "img" doivent obligatoirement **comporter un slash (/) à la fin** : `<br />`

Notez que l'espace entre le "br" et le slash n'est pas obligatoire, mais le W3C recommande de le mettre, alors en gentil garçon sage je le mets 😊

- Tous vos textes doivent être dans des balises de paragraphes. Il est interdit de mettre du texte directement entre les balises `<body></body>` sans l'avoir entouré des fameux `<p></p>`  
Ceci est aussi valable pour les retours à la ligne `<br />` et les images `<img />` : **ces balises doivent obligatoirement se trouver entre deux balises `<p></p>` !**

C'est une erreur ultra-courante chez les débutants. Voici un exemple de ce qu'il ne faut pas faire :

#### Code : HTML

```
<p>Ceci est un texte correctement placé dans un paragraphe.<br />
Les balises "br" doivent se trouver à l'intérieur d'un paragraphe, ne

Ceci est un texte en-dehors d'un paragraphe. C'est interdit.<br />
De même, les sauts de lignes "br" sont interdits en dehors d'un paragraphe.


<p>Ce texte est à nouveau dans une balise de paragraphe, là tout va bien.
N'oubliez pas que les images aussi doivent se trouver dans un paragraphe.

<p>
(Cette image est bien placée dans un paragraphe)</p>
```

- **Toutes vos images doivent comporter un attribut "alt"** qui indique ce que contient l'image :  
``  
 Si, par hasard, votre image est purement décorative (vous ne pouvez pas en trouver de description), vous êtes autorisés à ne rien mettre comme valeur pour l'attribut alt, comme ceci :  
``
- Vos balises doivent être **fermées dans l'ordre**. Je m'explique, voici ce qu'il ne faut pas faire :  
`<p>Texte <em>important</p></em>`  
 Maintenant, voici la bonne façon de faire :  
`<p>Texte <em>important</em></p>`  
 Gardez bien ce schéma en tête, beaucoup de débutants font cette erreur.
- Si vos liens comportent des `&` (c'est le cas des sites utilisant le langage PHP), vous devez taper le code `&amp;` à la place pour éviter une confusion au navigateur. Voici le mauvais exemple :  
`<a href="http://www.site.com/?jour=15&mois=10&an=2000">`  
 Voici le bon exemple, il m'a suffi de transformer les `&` en `&amp;` :  
`<a href="http://www.site.com/?jour=15&amp;mois=10&amp;an=2000">`
- Vérifiez enfin que vous n'avez utilisé que des balises XHTML et non pas de vieilles balises HTML. Si vous n'utilisez que les balises présentées dans ce cours, vous n'aurez aucun problème. Si vous avez utilisé des balises étrangères (comme `<marquee>`) ou obsolètes (comme le vieux `<frame>`), votre page contiendra des erreurs.  
 Le validateur vous dira "Element *bidule* undefined" ou encore "There is no attribute *truc*"

## Le validateur CSS

Là je vous rassure, ça va être plus rapide 😊

Tout d'abord, voici l'URL du validateur CSS à mettre en favoris :

<http://jigsaw.w3.org/css-validator/>

Vous devez indiquer l'adresse où se trouve le fichier CSS que vous voulez valider. Vous pouvez aussi, comme pour les pages XHTML, envoyer directement depuis votre disque dur vos CSS.

Notez que, comme pour les fichiers XHTML, si vous avez plusieurs fichiers CSS vous devez tous les valider.

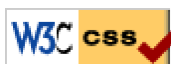
Vous n'êtes pas obligé d'indiquer le fichier CSS directement, vous pouvez donner l'url de votre page XHTML (.html) et le validateur ira directement chercher tous les fichiers CSS que vous utilisez (grâce à la balise `<link />` !)

Il faut savoir qu'il est nécessaire de passer d'abord par le validateur XHTML, de corriger vos erreurs, puis de passer au validateur CSS.

Le validateur CSS refusera probablement de se lancer si votre page XHTML comporte encore des erreurs.

Bon, en général la validation des CSS est une vraie partie de plaisir (ça change un peu du validateur XHTML :p). Pour avoir des erreurs, il faut que vous vous soyez complètement plantés quelque part ou que vous ayez utilisé des propriétés CSS non officielles. Voici les quelques rares erreurs que vous pouvez avoir fait :

- Vous avez oublié de **terminer une ligne d'une propriété CSS par un point-virgule (;)**. Si vos propriétés CSS ne se terminent pas par un point-virgule, votre fichier CSS est inexploitable.
- Vous avez oublié de **donner un nom de police standard à la fin de la propriété *font-family***. Pas bien :  
`font-family:Arial, "Trebuchet MS", Impact;`  
 Bien : `font-family:Arial, "Trebuchet MS", Impact, sans-serif;`
- Vous avez utilisé **une propriété CSS non officielle**. C'est le cas de Microsoft qui a inventé seul dans son coin des propriétés CSS qui ne marchent que sur Internet Explorer.  
 Parmi ces propriétés inventées, signalons `scrollbar-face-color` et compagnie (couleur de la barre de défilement) et les filtres comme `filter:glow()`. Ces propriétés n'étant pas officielles, vous ne devez pas les utiliser sur votre site web.



Vous avez réussi à créer des fichiers CSS valides ? Bravo ! (C'était pas bien dur :p)

Vous pouvez mettre le logo "Valide CSS" ci-contre pour montrer que vous utilisez des feuilles de styles qui respectent les normes. Là encore, rien ne vous empêche de créer le vôtre si vous trouvez que celui-ci est moche, ou même de ne rien mettre du tout si ça ne fait pas joli dans le design de votre site web.

J'espère que cette annexe vous a plu et que vous y avez appris de nouvelles choses intéressantes.

... Bon ok ça fait un peu bateau comme phrase de conclusion 😊

Plus sérieusement, c'est une annexe à ne pas prendre à la légère : j'ai essayé d'y résumer tous les enjeux du Web, son développement, son histoire. Une histoire dont VOUS allez faire partie désormais en tant que Webmaster.

Il est important que vos sites soient valides XHTML et CSS, que vous respectiez ces normes afin de pouvoir aider le Web à évoluer.

Et au-delà du respect des standards, c'est aussi le respect de l'internaute que vous devez avoir en tête. Mettre une petite publicité pour aider à financer votre site, ok. Mais noyer le visiteur de popups et javascripts en tous genres, s'il vous plaît non.

Nêtes vous pas parfois agacés sur certains sites un peu trop envahissants ? Mettez-vous à la place des internautes, créez le site que vous auriez envie de visiter.

Si vous respectez toutes ces consignes, vous n'avez pas à vous en faire : le succès de votre site suivra 😊

Ce n'est qu'une question de temps et de persévérance 😊