

# Introduction aux CSS

## Un cours complet sur les CSS.

Par Christophe F. 

Date de publication : 11 octobre 2011

Dernière mise à jour : 26 novembre 2011

Internet est devenu aujourd'hui un outil de communication extrêmement puissant, qui a pris une importance prépondérante dans notre société contemporaine. Selon les statistiques de juin 2010 il y aurait près de 2 milliards (1) d'utilisateurs d'internet dans le monde. Ce succès conduit logiquement à ce que de plus en plus de gens s'intéressent à la réalisation de sites web.

Pour créer un site web il faut s'atteler à l'apprentissage de langages tels que le HTML et le CSS. Les CSS, aussi appelées feuilles de style, sont devenues aujourd'hui, au même titre que le langage HTML, un maillon essentiel à la création de sites web. Elles permettent un large contrôle sur la présentation d'un document web.

Je vous propose une série d'articles qui va vous introduire dans le monde du langage CSS, en suivant différentes étapes vous permettant de vous familiariser petit à petit avec les feuilles de style. Chaque étape vous conduira à progresser dans la difficulté et dans l'approfondissement de l'étude du langage.

Pour enrichir l'exposé théorique, chaque article sera agrémenté de nombreux morceaux de code illustrant mes propos et qui renverront vers des pages d'exemple vous permettant de visualiser leur mise en œuvre pratique et vous donneront accès au code complet utilisé pour faire fonctionner l'exemple.

Vous aurez ainsi à votre disposition pour le moment **14** exemples sur un ensemble de **deux** articles de cette série.

Cet ensemble d'articles sur CSS est destiné autant au débutant qu'à celui qui connaît déjà un peu le CSS. Le premier y trouvera tout ce qu'il aura besoin de savoir pour bien débiter l'apprentissage. Le second pourra approfondir ses connaissances. Des connaissances du langage HTML sont cependant nécessaires.

**Dans ce premier article** nous allons aborder les points essentiels à retenir avant de se lancer dans l'étude du langage proprement dit. Nous allons d'abord vous indiquer ce que sont les CSS. Pour mettre en œuvre tout ce que vous allez apprendre il vous faudra des outils spécifiques, c'est le point que nous allons voir en second lieu. Dans le prolongement des outils, il faut savoir que les CSS sont régies par des "*normes*" rédigées par un organisme qui s'occupe de la gestion des CSS. Mais pourquoi utiliser les CSS ? Tout simplement parce qu'elles apportent des avantages que je vous détaillerai ensuite. Enfin, pour une meilleure compréhension de ces articles, j'aborderai les conventions utilisées dans tous ces articles.

Cet article est le premier d'une série consacrée à CSS, retrouvez les autres articles de cette série dans [le sommaire](#) ou dans l'encadré bleu [en bas de page](#).

Tous mes remerciements vont à **ClaudeLELOUP** pour sa relecture et ses corrections, ainsi qu'à **Macmillenium**, **Bovino** et tous les autres membres de **l'équipe Web** que j'ai oubliés, pour leur soutien.

Commentez cet article :

1.  **Source Wikipédia**

Introduction aux CSS.....	4
A. Qu'est-ce que les CSS.....	4
B. Les outils nécessaires pour utiliser le CSS.....	9
1. Outils d'édition du code.....	10
2. Les navigateurs.....	12
a. Trident : Internet Explorer.....	13
i. Internet Explorer 6.....	14
ii. Internet Explorer 7.....	14
iii. Internet Explorer 8.....	14
iv. Internet Explorer 9.....	15
b. Gecko : Firefox, Seamonkey, Camino et Netscape.....	15
c. Presto : Opera.....	16
d. KHTML et Webkit : Safari, Google Chrome et Konqueror.....	16
i. Konkeror.....	17
ii. Safari.....	17
iii. Google Chrome.....	17
C. La gestion du CSS.....	18
D. Avantages qu'apportent les feuilles de style à la création de sites web.....	19
1. Réduction de la quantité du code.....	19
2. Centralisation de la présentation.....	20
3. Adaptation du site en fonction de l'affichage.....	20
4. Création de feuilles de style personnalisées.....	20
E. Conventions et contenu de ce cours.....	21
1. Les exemples.....	22
2. Mise en avant des noms d'éléments dans le cours.....	23
3. Liste des articles liés.....	23
Remerciements.....	23

## Introduction aux CSS

Cette introduction va aborder les points essentiels à retenir avant de se lancer dans l'étude du langage proprement dit. Nous allons d'abord vous indiquer ce que sont les CSS. Pour mettre en œuvre tout ce que vous allez apprendre il vous faudra des outils spécifiques, c'est le point que nous allons voir en second lieu. Dans le prolongement des outils, il faut savoir que les CSS sont régies par des "normes" rédigées par un organisme qui s'occupe de la gestion des CSS. Mais pourquoi utiliser les CSS ? Tout simplement parce qu'elles apportent des avantages que je vous détaillerai ensuite. Enfin, pour une meilleure compréhension de ces articles, j'aborderai les conventions utilisées dans tous ces articles.

### A. Qu'est-ce que les CSS

Le langage CSS (en anglais Cascading Style Sheets, qu'on peut traduire par *feuilles de style en cascade*) a été créé pour le web : il sert principalement à gérer l'aspect visuel d'une page web (cette utilisation peut aussi se faire sur d'autres supports : XML, ODT, RSS, applications XUL <sup>(1)</sup> ...).

En d'autres termes, le langage CSS gère la présentation d'une page web : la couleur du texte, l'utilisation de telle ou telle police, les images ou couleurs d'arrière-plan, les espaces et marges des différents éléments, le positionnement des éléments, etc.

Le CSS n'existe pas par lui-même. Il s'applique forcément à un autre document : il ne peut être utilisé qu'en association avec des langages de balisage tels que HTML et XHTML.

Ces langages de balisage servent à définir une structure sémantique au contenu de la page web ; cela veut dire que les balises utilisées permettent de donner une signification à leur contenu : en indiquant que tel texte est un titre, tel autre une liste, ou encore tel autre un paragraphe. Le CSS est là pour gérer l'apparence de la page web, il crée la mise en page du langage de balisage : couleurs, positions, tailles, police... sans avoir à modifier le code (X)HTML. C'est un des concepts fondamentaux actuels du web : la séparation du fond et de la forme.

C'est-à-dire que le fond est géré par le langage de balisage (HTML) : si vous avez une liste vous utilisez les balises `<ul>` ou `<ol>` sans tenir compte de comment l'élément sera affiché. Car l'affichage relève de la forme qui sera gérée par les CSS. Vous pourrez spécifier le type de puce affiché, ou remplacer la puce par une image, déterminer la couleur du texte, etc. Ces deux couches que sont le fond et la forme doivent être séparées pour avoir des pages optimisées et bénéficier de toute la puissance du CSS. Nous verrons plus bas tous les avantages que va vous apporter ce langage.



*Il est donc important d'avoir déjà quelques notions de ces langages de balisage avant de se plonger dans l'apprentissage du CSS.*



*Vous pouvez vous référer à ce cours pour apprendre les bases du HTML : [les bases du html](#).*

Pour illustrer notre propos nous allons prendre un exemple :

#### Un exemple de code XHTML :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<title>Exemple 01 : Mise en page d'une structure XHTML sans CSS</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<meta http-equiv="Content-Style-Type" content="text/css" />
</head>

<body>
```

(1) Qu'est-ce que le ODT : [un format utilisé par Open Office](#)

Qu'est-ce que le XUL : [XML-based\\_User\\_interface\\_Language](#).

Un exemple de code XHTML :

```

<h1>
Exemple 01 : Mise en page d'une structure XHTML sans CSS
</h1>

  <div id="entete">
    
    <h2>
Exemple de site sans CSS
    </h2>
    <p>
Un exemple de présentation avec et sans code CSS.
    </p>
  </div>

  <div id="corps">
    <div id="menu">
      <h3>
Menu
      </h3>
      <ul>
        <li>
          <a href="#">
Lien 1
          </a>
        </li>
        <li>
          <a href="#">
Lien 2
          </a>
        </li>
        <li>
          <a href="#">
Lien 3
          </a>
        </li>
      </ul>
    </div>

    <div id="contenu">
      

      <h3>
Contenu
      </h3>

      <p>
Paragraphe 1
      </p>
      <p>
Paragraphe 2
      </p>
      <p>
Paragraphe 3
      </p>
      <p>
Paragraphe 4
      </p>
      <p>
Paragraphe 5
      </p>
    </div>

  </div>

  <div id="pied">
    <p>
Pied de page
    </p>
  </div>

</body>

```

## Un exemple de code XHTML :

```
</html>
```

Voyez ce que donne le code ci-dessus dans **l'exemple 01 sur une mise en page d'une structure XHTML sans CSS**. Cette page web n'est associée à aucun code CSS autre que celui de la feuille de style par défaut du navigateur. Cette feuille de style, qui est propre à chaque navigateur, ne fait que gérer le style basique des éléments du document. L'affichage se fait donc de manière brute, sans réelle mise en page autre que celle associée aux éléments (X)HTML.

Associons maintenant à ce code la feuille de style suivante :

## Un exemple de code CSS :

```
#entete, #corps, #pied {
    margin :      5px 0;
    padding :     2px;
    border :      1px solid black;
    background-color : #d3d3d3;/*lightgray*/
}

#entete {
    overflow :    hidden;
    height :      110px;
}

h1, h2, h3 {
    font-family : Arial, sans-serif;
    font-weight : bold;
    color :       blue;
    text-decoration : underline;
    text-align :  left;
}

#entete p {
    margin :      25px 0px;
}

#entete img {
    float :       right;
}

#corps {
    overflow :    hidden;
    zoom :        1;
}

#menu {
    width :       20%;
    float :       left;
}

#contenu {
    width :       80%;
    float :       left;
}

#contenu p:first-letter {
    font-weight : bold;
    font-size :   2.5em;
    background :  #add8e6;/*lightblue*/
    margin :      2px;
    padding :     4px;
    float :       left;
}

#contenu img {
    float :       right;
    margin :      0px 5px;
}

#pied {
```

## Un exemple de code CSS :

```
text-align : center;
}
```

Et voyons dans **l'exemple 02 sur une mise en page d'une structure XHTML avec CSS** ce que donne ce code CSS associé au document XHTML précédent. Notez que nous avons simplement rajouté la ligne suivante dans l'en-tête du document XHTML qui nous permet de lier le fichier CSS à la page XHTML (vous retrouverez le code complet dans l'exemple) :

## Rajout de la balise link pour lier le document CSS :

```
<link href="style-ex02.css" type="text/css" rel="stylesheet" />
```

En comparant les deux versions vous vous rendrez compte de l'immense potentiel du CSS. Cela vous donne également une vision sur les différents éléments gérés par le CSS.

Vous vous demandez certainement : « mais comment faisaient-ils avant que les CSS ne soient créées ? » Le plus simple est encore de l'illustrer par un exemple :

## Un exemple de site avec les éléments de présentation dans le code XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<title>Exemple 03 : Mise en page d'une structure XHTML avec l'ancienne méthode</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<meta http-equiv="Content-Style-Type" content="text/css" />
</head>

<body>

<h1>
Exemple 03 : Mise en page d'une structure XHTML avec l'ancienne méthode
</h1>

<table border="0" width="100%">
<tr>
<td>

<table border="1" width="100%" bgcolor="lightgrey" cellspacing="0">
<tr>
<td>

<table border="0" width="100%" cellspacing="0">
<tr>
<td align="left">
<h2>
<font face="Arial, Serif" color="blue">
<u>
Exemple de site avec CSS
</u>
</font>
</h2>
<p>
Un exemple de présentation avec tableaux.
</p>
</td>
<td align="right">

</td>
</tr>
</table>

</td>
</tr>
</table>
```

## Un exemple de site avec les éléments de présentation dans le code XHTML

```

        </td>
    </tr>
    <tr>
        <td height="1">
        </td>
    </tr>
    <tr>
        <td>

        <table border="1" width="100%" bgcolor="lightgrey" cellspacing="0">
            <tr>
                <td>

                <table border="0" width="100%" cellspacing="0">
                    <tr>
                        <td width="20%" valign="top">
                            <h3>
                                <font face="Arial, Serif" color="blue">
                                    <u>
Menu
                                </u>
                                </font>
                            </h3>
                            <ul>
                                <li>
Lien 1
                                    <a href="#">
                                        </a>
                                </li>
                                <li>
Lien 2
                                    <a href="#">
                                        </a>
                                </li>
                                <li>
Lien 3
                                    <a href="#">
                                        </a>
                                </li>
                            </ul>
                        </td>
                        <td width="80%">
                            

                            <h3>
                                <font face="Arial, Serif" color="blue">
                                    <u>
Contenu
                                </u>
                                </font>
                            </h3>

                            <p>
<font size="7"><b>P</b></font>aragraphe 1
                            </p>
                            <p>
<font size="7"><b>P</b></font>aragraphe 2
                            </p>
                            <p>
<font size="7"><b>P</b></font>aragraphe 3
                            </p>
                            <p>
<font size="7"><b>P</b></font>aragraphe 4
                            </p>
                            <p>
<font size="7"><b>P</b></font>aragraphe 5
                            </p>
                        </td>
                    </tr>
                </table>
            </tr>
        </table>
    
```

### Un exemple de site avec les éléments de présentation dans le code XHTML

```

        </td>
    </tr>
</table>

</td>
</tr>
<tr>
    <td height="1">
    </td>
</tr>
<tr>
    <td>

        <table border="1" width="100%" bgcolor="lightgrey" cellspacing="0">
            <tr>
                <td align="center" height="50">
Pied de page
            </td>
            </tr>
        </table>

    </td>
</tr>
</table>

</body>
</html>
    
```

Ce code va aboutir à l'affichage suivant : visualiser le résultat dans **l'exemple 03 : Mise en page d'une structure XHTML avec l'ancienne méthode des tableaux**. Vous constaterez qu'il n'y a pas d'énormes différences d'affichage entre cette version et celle ci-dessus. La différence fondamentale réside dans le code : la mise en page est faite à base de tableaux, tous les aspects de la présentation sont gérés directement dans le code XHTML et à l'aide d'attributs relevant de ce langage.

 Vous pouvez cependant noter que pour garder une page valide nous avons dû modifier le DOCTYPE utilisé : nous sommes passés du DOCTYPE Strict au DOCTYPE Transitional. En effet les attributs de présentation dans le code (X)HTML ne sont pas admis avec un DOCTYPE Strict, tandis qu'ils sont autorisés si on utilise un DOCTYPE Transitional.

Mais pourquoi alors s'embêter avec un nouveau langage si le HTML s'autosuffit ? Parce que le CSS est un énorme progrès et qu'il a de nombreux avantages tels que légèreté du code, facilité de changer de design... Le code de l'exemple 3 est moins pratique, moins maintenable et plus lourd.

Aujourd'hui le (X)HTML ne peut plus se passer du CSS. Pour se conformer aux bonnes pratiques en la matière, il est en effet déconseillé de faire des mises en page à l'aide de tableaux <sup>(2)</sup> ni d'intégrer des attributs de présentation au sein du code (X)HTML. Il faut séparer au maximum la structure du document de sa mise en forme, et le CSS est là pour répondre à ce besoin.

Il permet surtout de faire des choses qui ne sont pas possibles ou, pour le moins, beaucoup plus complexes à réaliser en HTML pur.

Je détaillerai les avantages qu'apporte le CSS [un peu plus loin](#) dans cet article.

## B. Les outils nécessaires pour utiliser le CSS

Pour créer un site web il vous suffit de créer deux fichiers : le premier avec l'extension .htm ou .html, et le second avec l'extension .css correspondant au fichier CSS.

(2) Pourquoi est-il déconseillé de faire des mises en page à l'aide de tableaux : [FAQ FAQ HTML-XHTML](#).

Une fois ces fichiers créés il vous faudra deux outils principaux pour réaliser un site web : un outil permettant d'éditer ces fichiers et un navigateur pour pouvoir visualiser le résultat.

## 1. Outils d'édition du code

Ces deux types de fichiers (.html et .css) sont de simples fichiers texte qui sont modifiables avec un éditeur de texte basique, dans lesquels vous allez écrire du texte formaté d'une certaine manière. Un outil évolué n'est pas nécessaire pour ce faire.

Au niveau de ces outils vous avez le choix parmi une pléthore de logiciels, des plus simples aux plus évolués, certains nécessitant de votre part une connaissance du code, d'autres vous permettant de réaliser un site web sans presque aucune connaissance des langages employés. Ce ne sont bien sûr pas ces derniers que nous allons privilégier ici puisqu'il s'agit de vous initier à ce langage qu'est le CSS.

Un logiciel extrêmement rudimentaire comme le Bloc-notes de Windows va suffire pour faire du CSS. Vous trouverez ce logiciel en cliquant sur *démarrer > Tous les programmes > Accessoires > Bloc-notes*. Pour Mac vous disposez de TextEdit que vous trouverez dans le dossier *Application*. En ce qui concerne Linux les différentes distributions existantes sont trop nombreuses pour indiquer tous les éditeurs disponibles, on peut par exemple citer nedit, gedit, emacs, xemacs, etc.

 *Si vous enregistrez vos documents avec l'un de ces logiciels, en particulier avec le Bloc-notes de Windows, prenez soin de bien spécifier la bonne extension. En effet ces logiciels vont enregistrer par défaut votre document au format .txt. Si vous n'indiquez pas explicitement l'extension .css votre fichier ne sera pas reconnu comme tel et n'aura pas l'effet désiré.*

Le Bloc-notes de Windows est un des logiciels les plus basiques qui existent pour écrire du code CSS ou (X)HTML. Il en est d'autres qui disposent de fonctionnalités qui vont assister le développeur dans sa tâche.

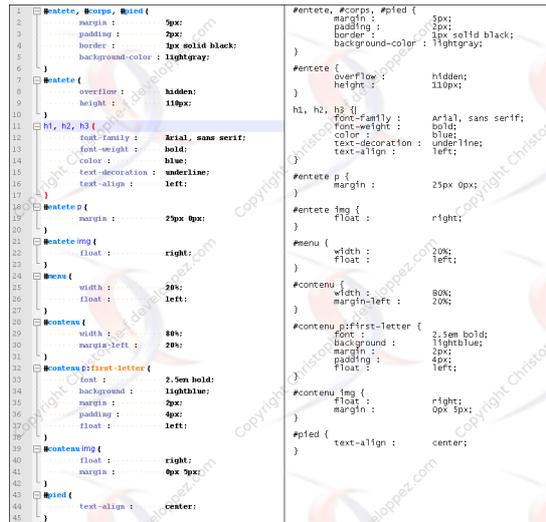
Ces outils peuvent être classés en deux catégories :  **les éditeurs de texte** et les  **EDI** (environnement de développement intégré, de l'anglais IDE (Integrated Development Environment) <sup>(3)</sup> .

Ces outils partagent tous un certain nombre de fonctionnalités utiles au développeur. Une des fonctionnalités les plus pratiques est la coloration syntaxique : le logiciel affiche le code en couleurs en fonction des différents éléments qui le composent. Cela permet une lecture plus rapide du code ainsi qu'une visualisation simplifiée des erreurs. En fonction des logiciels, d'autres fonctionnalités sont proposées comme l'autocomplétion (le fait, au fur et à mesure que vous tapez le code, de proposer les différents mots-clés, vous évitant ainsi de les taper entièrement), la fonction avancée de Rechercher/Remplacer, etc.

Les EDI sont cependant plus perfectionnés que les éditeurs de texte, car ils proposent d'autres outils encore plus évolués tels qu'un débogueur, un compilateur (pour les langages compilés), une interface de gestion de versions, etc.

---

(3) Voir aussi  [la définition sur Wikipédia](#).



*Comparaison entre Notepad++ et le Bloc-notes de Windows : l'un possédant la coloration syntaxique, l'autre pas*

Vous trouverez différentes rubriques listant ces logiciels sur le site [developpez.com](http://xhtml.developpez.com/outils/), à l'adresse suivante <http://xhtml.developpez.com/outils/> pour les outils (X)HTML et à l'adresse suivante <http://css.developpez.com/outils/> pour les outils CSS.

Pour ma part, et c'est un avis totalement personnel, je vous conseille **Notepad++** qui est un éditeur de texte et non un EDI. Parmi les fonctionnalités dont dispose cet outil je citerai celles-ci :

- quand vous positionnez votre curseur sur une balise double, une accolade ou des parenthèses, ces éléments sont surlignés ou changent de couleur. Cela permet de les distinguer parmi les autres, de savoir quel élément fermant est associé à celui qu'on est en train de visualiser. Bref cela permet d'avoir une meilleure vision du code ;
- devant chaque balise vous disposez d'un contrôle vous permettant, en cliquant dessus, de cacher ou d'afficher la portion de code (visible sur la photo ci-dessus à gauche) ;
- vous disposez de deux vues positionnées soit verticalement soit horizontalement, vous permettant de comparer deux documents. Vous avez aussi des contrôles vous permettant de synchroniser soit le scroll vertical, soit le scroll horizontal, soit les deux. Un outil existe également permettant de mettre en avant les différences dans le contenu de chaque fenêtre ;
- Notepad++ gère un large panel de langages, tous associés à leur coloration syntaxique, que vous pouvez personnaliser à votre guise.

Mais il existe plein d'autres logiciels, à vous de faire votre choix.

Je ne peux pas aborder les outils pour les développeurs web sans dire un mot sur les outils WYSIWYG, même si le but de ce cours est de vous apprendre le CSS et donc de vous faire plonger directement dans le code CSS. En effet ces outils permettent à un néophyte de créer un site web sans aucune connaissance des langages qui régissent le web.

Que signifie WYSIWYG ? Ce sigle signifie What You See is What You Get, traduit en français : ce que vous voyez est ce que vous obtenez. Ce genre de logiciel permet de créer des sites web en mode graphique, sans taper une seule ligne de code, un peu à la mode des logiciels de traitement de texte pour la mise en page du texte.

Ce type de logiciel est donc utile pour des gens qui ne connaissent pas les langages du web. Ils peuvent créer un site à partir de zéro sans taper une seule ligne de code. Le principal inconvénient est que vous n'avez pas la main sur le code, le logiciel le créant lui-même en fonction de vos actions dans l'interface graphique de celui-ci. Ce code n'est pas toujours propre et ne respecte pas les standards tels qu'élaborés par le W3C.

Il faut cependant préciser que certains de ces logiciels disposent également d'un mode d'édition du code : vous pouvez visualiser le code et le modifier à votre guise, comme dans un simple éditeur de texte. Parmi les logiciels à conseiller vous trouverez  **Dreamweaver** (payant) ou  **Komposer** (gratuit, successeur de  **Nvu**).

Une fois vos premières lignes de code écrites, vous voudrez certainement visualiser le résultat, pour cela il vous faut un navigateur.

## 2. Les navigateurs

Une page web est destinée, en principe, à être visualisée par le plus grand nombre ! Pour ce faire il existe des logiciels qu'on appelle des navigateurs graphiques par opposition aux navigateurs textuels en mode console et aux logiciels de synthèse vocale.

Les navigateurs textuels en mode console affichent un site en mode texte uniquement, sans appliquer le code CSS et sans télécharger les images (exemple : Lynx). Les logiciels de synthèse vocale sont destinés aux malvoyants (ils convertissent le texte en voix de synthèse), notez que le CSS dispose de propriétés permettant de contrôler la voix générée par ces navigateurs vocaux <sup>(4)</sup> (exemple : [Opera](#) à partir de sa version 7).



*Il ne sera ici question que des navigateurs graphiques.*

Un navigateur graphique est composé de deux éléments principaux : un moteur de rendu et une interface graphique.

Le moteur de rendu sert à décrypter les langages du web, (X)HTML et CSS entre autres, pour les convertir en une représentation graphique dans le navigateur de l'utilisateur, donc avec la mise en forme et tous les éléments de design <sup>(5)</sup>. L'interface graphique est une représentation graphique sur un écran d'ordinateur permettant une interaction de la part de l'utilisateur à l'aide de périphériques tels que souris, clavier...

Les navigateurs sont le talon d'Achille de CSS. En effet, ces logiciels sont la propriété de sociétés commerciales pour la plupart, qui sont libres de leurs choix de développement, je veux dire par là que les spécifications CSS que nous allons aborder dans la suite de cet article, n'ont aucune valeur contraignante et que ces éditeurs de logiciels choisissent de les implémenter ou non, et de la manière qui leur convient.

Donc ces spécifications CSS, qui pourraient être le socle commun sur lequel le développeur web devrait pouvoir se reposer, sont implémentées de manière inégale dans les différents navigateurs existant sur le marché. Certains implémentent les différentes règles de manière conforme aux spécifications, d'autres les implémentent incorrectement voire pas du tout.

Il faut donc composer avec ces derniers pour créer un site web qui sera affiché uniformément sur tous les navigateurs. Pour cela il n'y a pas d'autre solution que de tester au maximum son site web avec les navigateurs les plus fréquemment utilisés et qui représentent la plus large part de marché possible <sup>(6)</sup>. Il est en effet difficilement concevable d'exclure une partie des utilisateurs qui disposent de tel ou tel navigateur au seul motif que le développeur n'a pas su ou voulu adapter son site à ce navigateur, d'autant plus lorsque les parts de marché de ce(s) navigateur(s) sont importantes.

Il restera cependant toujours une part d'aléas : quand les spécifications n'ont prévu aucune règle pour tel ou tel point, alors les programmeurs de ces logiciels vont mettre en œuvre des solutions qui peuvent leur être propres. Donc même avec les navigateurs respectueux des standards vous pourrez rencontrer des différences d'affichage sur ce genre de point.

<sup>(4)</sup> Ces propriétés ne seront pas abordées ici :  **Spécifications CSS 2** et  **Module Speech des spécifications CSS 3**.

<sup>(5)</sup> Liste des principaux moteurs de rendu actuels :  [http://fr.wikipedia.org/wiki/Moteur\\_de\\_rendu\\_de\\_pages\\_web](http://fr.wikipedia.org/wiki/Moteur_de_rendu_de_pages_web)

<sup>(6)</sup> Voici un article sur le sujet :  **Tester ses navigateurs de façon fiable**.

De plus les navigateurs ont tous une feuille de style par défaut pour tous les éléments (X)HTML dont les règles peuvent varier en fonction des navigateurs.

Certains développeurs ont créé des tests permettant de vérifier le respect des standards par les différents navigateurs. Il existe en particulier le test ACID décliné en trois versions <sup>(7)</sup>. Les tests ACID sont destinés à vérifier que les moteurs de rendu respectent un certain nombre de standards du web. Je préciserai pour chaque navigateur quel résultat il obtient au test <sup>(8)</sup>. Les niveaux se calculent avec une note pouvant aller de 0 à 100.

Maintenant que nous avons planté le décor, il nous reste à évoquer les principaux navigateurs qu'il convient de prendre en compte. Nous allons les répartir en fonction de leur moteur de rendu, c'est lui qui gère la présentation, et donc des navigateurs disposant du même moteur de rendu vont avoir un affichage relativement similaire.

## a. Trident : Internet Explorer

Internet Explorer a été le navigateur de référence pendant longtemps. Il a en effet atteint un pic de 95 % de parts de marché au début des années 2000 <sup>(9)</sup>. Son utilisation, par les internautes, est aujourd'hui en constant déclin, mais reste encore significative. Avec entre 40 et 55 % de parts de marché, selon les sources, il est encore le navigateur le plus répandu.

Internet Explorer est un navigateur qui n'est pas très respectueux des standards et qui peut poser de nombreux soucis de rendu dans vos pages web. Cette situation est cependant sur le point d'évoluer avec les dernières versions.

Parmi les particularités d'Internet Explorer on trouve le concept de layout (aussi appelé haslayout), le modèle de boîte spécifique de Microsoft qui diffère du modèle standard du W3C et le mode de compatibilité descendante quirks qui est un peu plus présent dans Internet Explorer que dans les autres navigateurs. Les concepts de modèle de boîte Windows et le mode de compatibilité quirks, n'étant pas spécifiques à Internet Explorer, seront étudiés dans un article ultérieur.

Le concept de haslayout est un concept propre à Internet Explorer. C'est un mécanisme spécifique qui existe dans le moteur de rendu dans les versions 5 à 7 d'Internet Explorer, et qui influe sur l'affichage et le positionnement des éléments, ainsi que leur interaction avec les autres éléments.

Dans la pratique, le layout peut avoir des effets positifs comme des effets négatifs : parfois un élément ne s'affichera correctement que s'il a le layout ; dans d'autres cas si l'élément a le layout, il ne s'affichera pas correctement et vous serez confronté à un bogue de rendu.

Le layout n'est pas une propriété CSS, il ne peut donc pas être conféré via une propriété 'layout'. En fait le layout est une qualité qui est ou non attribuée aux éléments. Certains éléments (X)HTML ont par défaut cette qualité. Pour que les autres éléments aient le layout, il faut leur appliquer une des propriétés CSS qui a le pouvoir de conférer le layout (ces propriétés sont entre autres **width**, **height**, **zoom**, **float**, etc.). Appliquer une de ces propriétés à un élément lui confère alors une sorte de valeur haslayout = true. Pour mettre en œuvre la valeur haslayout = false il n'y a que la possibilité de supprimer la propriété CSS qui a conféré le layout.

Pour plus d'informations sur le concept de layout vous pouvez consulter les articles suivants : [Avoir le layout - Le concept de hasLayout dans IE/Win](#) et [Avoir la mise en page](#).

Le moteur de rendu qu'il utilise est appelé Trident. Quatre versions d'Internet Explorer sont aujourd'hui toujours utilisées : ce sont les versions 6, 7, 8 et 9.

(7) Précisions sur les tests ACID : [ACID 1](#), [ACID 2](#), [ACID 3](#)

(8) Résultats des tests ACID 3 pour les différents navigateurs : [résultats des tests ACID 3 pour les navigateurs](#)

(9) Retrouvez ces statistiques sur Wikipédia : [http://en.wikipedia.org/wiki/Usage\\_share\\_of\\_web\\_browsers](http://en.wikipedia.org/wiki/Usage_share_of_web_browsers)

## i. Internet Explorer 6

Internet Explorer 6 est la version la plus ancienne de celles encore utilisées. Elle est sortie en août 2001. C'est un navigateur encore très répandu avec environ 15 % de parts de marché en 2010 <sup>(10)</sup>.

C'est un navigateur qui pose un certain nombre de problèmes de compatibilité avec les spécifications CSS : soit parce qu'il ne respecte pas ou partiellement certaines parties des spécifications, soit qu'il n'implémente pas ou mal certaines propriétés CSS. Il faudra cependant encore gérer ce navigateur pendant quelque temps au vu des parts de marché qu'il représente toujours.

Malgré cela, certains sites parmi les plus importants (notamment YouTube et Facebook) ont abandonné en 2010 le support de ce navigateur.

Ce navigateur fait un score de 12 sur 100 au test Acid 3.

## ii. Internet Explorer 7

Internet Explorer 7 est sorti en octobre 2006, soit cinq années après Internet Explorer 6. Par rapport à son prédécesseur il a fait l'objet de nombreuses améliorations du point de vue de la sécurité (filtre antiphishing...), des fonctionnalités (onglets...) et au niveau de la prise en charge des spécifications CSS 2.1.

Malgré les efforts consentis depuis Internet Explorer 6 au niveau du respect des standards, Internet Explorer 7 n'atteint pas encore le niveau de ses concurrents. Il reste le navigateur le moins respectueux des standards parmi les navigateurs modernes.

Ce navigateur fait un score de 12 sur 100 au test Acid 3.

## iii. Internet Explorer 8

Internet Explorer 8 est sorti le 19 mars 2009.

Cette version apporte un certain progrès dans le sens d'un plus grand respect des normes et standards du W3C : Internet Explorer 8 respecte presque totalement CSS 2.1. De même, le Haslayout, notion spécifique à Internet Explorer, disparaît. Par contre il ne va pas aussi loin que ses concurrents : il n'implémente presque aucun élément des spécifications CSS 3. Il n'implémente par exemple aucun module de CSS 3 et ce y compris les media queries <sup>(11)</sup>. Et n'implémente pas non plus le type Mime **application/xhtml+xml** : il n'est donc pas possible de faire du 'vrai' XHTML, c'est-à-dire auquel on applique les règles du XML, car le document sera interprété suivant les règles du HTML <sup>(12)</sup>.

Cependant cette meilleure prise en charge des standards du web va conduire à ce que les sites compatibles avec les anciennes versions d'Internet Explorer ne le sont plus avec Internet Explorer 8. De ce fait Microsoft a imaginé un nouveau système permettant de choisir le mode de compatibilité désiré : Internet Explorer 8 peut simuler le moteur de rendu d'Internet Explorer 7. En d'autres termes Internet Explorer 8 peut afficher une page web comme l'affichait Internet Explorer 7.

En réalité Internet Explorer 8 dispose de trois modes de compatibilité : le mode de compatibilité descendant quirks (commun à tous les navigateurs), le mode d'Internet Explorer 7 appelé mode de compatibilité, et enfin le mode standard (également commun à tous les navigateurs).

<sup>(10)</sup>Source Wikipédia :  [http://en.wikipedia.org/wiki/Internet\\_Explorer\\_6#Market\\_share](http://en.wikipedia.org/wiki/Internet_Explorer_6#Market_share) et  [http://en.wikipedia.org/wiki/Internet\\_Explorer#Market\\_share\\_history\\_overview\\_by\\_year\\_and\\_version](http://en.wikipedia.org/wiki/Internet_Explorer#Market_share_history_overview_by_year_and_version).

<sup>(11)</sup>Pour plus de renseignements sur les média queries :  [Le module CSS 3 sur les média queries](#)

<sup>(12)</sup>Pour plus de renseignements sur ces notions :  [comment servir du XHTML en tant que application/xhtml+xml](#).

Par défaut c'est le mode d'Internet Explorer 8 qui est utilisé. Le mode d'Internet Explorer 7 peut être activé soit à partir du navigateur, donc par l'utilisateur, soit à partir du site internet lui-même, donc par le webmaster.

Pour changer le mode de compatibilité à partir du site il faut envoyer par entête HTTP la propriété "X-UA-Compatible". Cela peut se faire par un entête serveur, un header PHP, ou encore par une balise meta :

Balise meta permettant de changer le mode de compatibilité d'Internet Explorer 8

```
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
```

 Plus de renseignements sur ces modes de compatibilité : [🇫🇷 tout savoir sur les modes de compatibilité d'Internet Explorer 8.](#)

Ce navigateur passe le test Acid 2 mais fait un score de 20 sur 100 au test Acid 3.

#### iv. Internet Explorer 9

Internet Explorer 9 est sorti le 14 mars 2011.

Il poursuit le progrès effectué par Microsoft dans le sens d'un meilleur respect des standards. Cette version introduit également HTML 5 et CSS 3 pour lesquels il fournit une compatibilité partielle.

Internet Explorer 9 se prévaut aussi d'une rapidité accrue dans l'affichage des pages notamment grâce à l'utilisation de l'accélération matérielle, ainsi qu'une meilleure vitesse d'exécution du JavaScript. Malheureusement ces améliorations rendent Internet Explorer 9 incompatible avec le système d'exploitation Windows XP.

Cette version d'Internet Explorer fait un score de 95/100 au test Acid 3, démontrant ainsi les progrès faits dans l'intégration des standards du web.

#### b. Gecko : Firefox, Seamonkey, Camino et Netscape...

Le premier navigateur commercial au milieu des années 1990 a été Netscape Navigator. Avec l'arrivée d'Internet Explorer il y a eu une « guerre des navigateurs » <sup>(13)</sup> qui a abouti à la défaite de Netscape. En effet ce navigateur a progressivement perdu de plus en plus de parts de marché, à tel point qu'Internet Explorer est devenu le seul navigateur de référence. Cette situation a décidé la société Netscape à rendre en 1998 son navigateur gratuit et à en libérer le code source.

C'est de là qu'est né le projet Mozilla qui était chargé du développement et de la gestion de ce code source. Mais après avoir constaté que ce code source était un obstacle au développement d'un nouveau navigateur, l'équipe de développeurs a décidé de repartir de zéro. De leurs travaux est né le nouveau moteur de rendu appelé Gecko.

Ce moteur de rendu est utilisé dans de nombreux navigateurs dont le plus connu est Mozilla Firefox, mais aussi Camino, Netscape versions 6 à 9, Mozilla Suite devenu SeaMonkey... Tous ces navigateurs sont supposés avoir le même comportement puisqu'ils utilisent le même moteur de rendu.

Mozilla Firefox est actuellement le concurrent le plus sérieux d'Internet Explorer ; il fait entre 20 et 35 % de parts de marché. Il a fait l'objet de nombreuses innovations techniques et surtout, ce navigateur est connu pour son respect toujours constant des normes du W3C.

Un autre avantage de Firefox est son système de modules : on peut rajouter tout un ensemble de fonctionnalités supplémentaires. Parmi ces modules il en existe quelques-uns qui aident le développeur dans la réalisation et le débogage du code (X)HTML et CSS : c'est par exemple le cas de  **Web Developer**, ou encore du module Firebug

<sup>(13)</sup>Voir cet article pour plus de renseignements : [🇫🇷 la guerre des navigateurs.](#)

dont vous trouverez un article décrivant ses fonctionnalités ici : [Maîtriser Firebug, l'indispensable extension Firefox pour le développement web.](#)

Firefox en est actuellement à sa version 8. Mais ce numéro va s'incrémenter rapidement depuis que Mozilla a décidé de modifier **son cycle de développement**. Une nouvelle version devrait voir le jour toutes les six semaines !

Firefox a sans cesse évolué au niveau de l'implémentation des standards : il est passé d'un score de 53 sur 100 pour Firefox 2 à un score de 71 puis 93 sur 100 pour Firefox 3 et Firefox 3.5. Il fait actuellement un score de 95 sur 100 pour ses deux dernières versions que sont Firefox 4 et 5.

### c. Presto : Opera

Opera est un navigateur norvégien. Il est connu pour sa rapidité et sa conformité aux standards. Le navigateur est à sa version 11.50 depuis juin 2011.

Ce navigateur est né en 1994. Il a été créé par des programmeurs de la société Televerket (aujourd'hui Telenor). C'est une compagnie publique de télécom norvégienne. La société Opera Software a été créée l'année suivante, soit en 1995, pour permettre un développement rapide du navigateur et de sa commercialisation. En effet ce navigateur n'est totalement gratuit que depuis 2005 dans sa version 8.5, puisqu'auparavant le navigateur intégrait un bandeau publicitaire.

C'est un navigateur qui est beaucoup moins utilisé que les précédents : il fait entre 1 et 2 % de parts de marché. Par contre il est très présent dans les appareils portatifs : téléphones portables et autres petits appareils tels que PDA... Mais aussi dans la Nintendo DS et la Wii... Son moteur de rendu est aussi, par exemple, utilisé dans Dreamweaver.

Opera est le seul navigateur parmi ceux cités ici qui fait aussi office de navigateur vocal : il permet de transformer le texte de la page web qui sera récité par un synthétiseur vocal. Il peut aussi être contrôlé à la voix. Opera supporte donc le module Speech de CSS 3. Cela permet aux aveugles de naviguer sur les sites web.

Parmi les fonctionnalités principales, Opera dispose de la navigation par onglets mais à la différence des autres navigateurs, les onglets d'Opera peuvent être déplacés, redimensionnés, empilés et affichés en cascade comme un système multifenêtre. À l'avant-garde au niveau des appareils portatifs, Opera dispose d'un mode qui permet de visualiser une page en taille réduite, donc de simuler un écran d'appareil mobile <sup>(14)</sup> (téléphone, PDA...).

Ce navigateur depuis sa version 10, fait un score de 100 sur 100 au tes Acid 3. L'ancienne version (Opera 9.64) faisait, elle un score de 85 sur 100.

### d. KHTML et Webkit : Safari, Google Chrome et Konqueror

KHTML est un moteur de rendu créé par et pour KDE (une des distributions de Linux). La première version est sortie en 2000.

C'est un moteur de rendu reconnu pour sa rapidité. Il respecte les normes HTML 4.1, CSS 1, presque entièrement CSS 2.1 (partiellement la partie sur les types de média), et certaines règles CSS 3 (particulièrement le module sur les sélecteurs).

Il faut aussi parler de Webkit qui est un moteur de rendu basé sur KHTML. Webkit a été créé par les développeurs d'Apple à partir du code de KHTML. Il est sorti pour la première fois en janvier 2003. Son code est devenu open source en juin 2005. C'est aussi un moteur de rendu qui est très respectueux des normes du W3C.

---

<sup>(14)</sup>Cette fonctionnalité a, semble-t-il, disparu avec la version 10.50.

Ces moteurs de rendu sont utilisés dans de nombreux outils, notamment grâce à Webkit qui permet d'intégrer son moteur de rendu appelé Webcore, très simplement dans n'importe quel programme. Trois des principaux navigateurs utilisent ces moteurs de rendu, ce sont entre autres Konkeror (Linux), Safari (Apple) et Google Chrome.

## i. Konkeror

Konkeror a été développé pour KDE (Linux). Il est complètement intégré à son interface puisque c'est non seulement un navigateur web mais aussi un gestionnaire de fichiers. Son interface utilisateur est largement configurable : il est possible de diviser la zone d'affichage, permettant de visualiser plusieurs dossiers locaux et distants, le tout imbriqué. Il dispose de tous les progrès des navigateurs actuels.

Ce navigateur est actuellement à sa version 4.7. Il fait un score de 89 sur 100 au test Acid 3 (version 4.5).

## ii. Safari

Les premiers systèmes d'exploitation Mac ne disposaient pas de navigateur propre : Mac OS était distribué, d'abord uniquement avec Netscape jusqu'en 1997, puis en rajoutant Internet Explorer Mac après 1997. En 2003 Microsoft a annoncé l'abandon du développement de son navigateur pour Mac.

Constatant cet abandon par Microsoft, Apple s'est, à la même époque, lancé dans le développement de son propre navigateur. Il a repris KHTML en l'adaptant, pour devenir le moteur de rendu Webcore. La première version de Safari a été présentée au public en janvier 2003.

Comme les autres navigateurs alternatifs, Safari a une bonne prise en charge des standards du W3C. C'est également un navigateur très rapide, la version 3 a apporté dans ce domaine un progrès par rapport à son prédécesseur...

En 2007 Apple a sorti une version de Safari pour Windows XP et Vista, qui était le navigateur le plus rapide disponible sur Windows à l'époque.

Le 8 juin 2009 était sorti officiellement Safari 4. Ce fut un des premiers navigateurs à avoir passé avec succès le test Acid 3. Cela démontre d'ailleurs l'intérêt qu'apporte Apple au respect des standards : déjà Safari 3 était le premier, en avril 2005, à passer avec succès le test Acid 2.

Safari 5, sorti en juin 2010, fait un score de 100 sur 100 au test Acid 3.

## iii. Google Chrome

Comme son nom l'indique c'est le navigateur de Google, le célèbre moteur de recherche. Google a annoncé le 1<sup>er</sup> septembre 2008 la naissance d'un nouveau navigateur : Google Chrome. La première version stable date du 11 décembre 2008.

Google Chrome utilise Webkit comme moteur de rendu. C'est un navigateur reconnu pour sa rapidité et son respect des standards.

C'est le navigateur qui a actuellement la plus forte marge de progression en termes de taux d'utilisation. Il est devenu en moins de trois ans le troisième navigateur avec un taux d'utilisation oscillant entre 12 et 16 % de parts de marché.

Google sort une nouvelle version de son navigateur environ toutes les six semaines, ce qui explique que Google Chrome en soit déjà à la version 14 qui est sortie en septembre 2011.

La première version de Chrome passait le test Acid 1, mais pas Acid 2, il faisait un total de 79 sur 100 points au test Acid 3. Depuis la version 4, google chrome fait un score de 100 sur 100 à ce test.

## C. La gestion du CSS

Le CSS est géré par le W3C (World Wide Web Consortium). Le W3C est un groupe créé en 1994 et qui a pour objectif de définir les langages du Web comme le HTML, le XHTML, le CSS, le XML...

Il a pour mission d'élaborer des standards sur le fonctionnement du web et sur les langages décrits à l'instant. Les documents que le W3C crée sont appelés spécifications, ces documents ont différents statuts en fonction de leur état d'avancement : le statut de *version de travail* (working draft) concerne les spécifications qui sont en cours d'élaboration, qui peuvent encore être sujettes à des modifications. On passe ensuite par une seconde étape appelée *dernier appel* (Last Call Working Draft), la troisième étape appelée *candidat à la recommandation* (candidate recommendation), est attribuée aux spécifications que le groupe de travail estime achevées et prêtes à être testées. Juste avant de passer à la dernière étape, le document de travail passe par une phase appelée *recommandation proposée* (Proposed Recommendation). Enfin, si la spécification est testée et ratifiée par le W3C, elle prend le statut de *recommandation* (recommendation).

Le W3C est composé d'entreprises et d'organisations qui sont concernées par le monde du Web, telles que Microsoft, Apple, la fondation Mozilla, Opera Software, Adobe, IBM, AOL, Google, HP, Sun Microsystems...

Les recommandations n'ont pas de caractère obligatoire, le W3C ne peut pas contrôler la mise en œuvre de ces standards. Il n'a pas ce pouvoir parce que ce sont des sociétés commerciales ou des organisations indépendantes qui vont mettre en œuvre ces standards dans leurs logiciels.

Ces programmes s'appellent "agent utilisateur". Un agent utilisateur est un programme qui met en œuvre une spécification du W3C. Plus spécifiquement au niveau du web, ce sont les navigateurs, mais pas seulement.

Les recommandations ont un intérêt essentiel, celui de permettre une interopérabilité des différents agents utilisateurs et systèmes d'exploitation. En d'autres termes cela permet d'avoir le même comportement et le même affichage quels que soient l'agent utilisateur et le système d'exploitation utilisé. S'il n'y avait pas de norme dans ce domaine, chaque agent utilisateur aurait ses propres normes et modalités d'affichage. Il serait donc nécessaire pour le développeur web de proposer des documents différents pour chaque agent utilisateur.

Il existe actuellement quatre versions des standards CSS : la première version du CSS (CSS 1) est devenue une recommandation en 1996, CSS 2 est devenue une recommandation en 1998, CSS 2.1 après être restée très longtemps en phase de *candidate recommendation*, est passée en phase de recommandation le 7 juin 2011. Enfin CSS 3 est en cours de rédaction, c'est une version de travail.

- La spécification CSS 1 a été achevée en décembre 1996. Elle proposait à l'époque une cinquantaine de propriétés qui avaient essentiellement pour objectif de gérer la mise en forme (police, couleurs, taille du texte...). Cette spécification n'aborde alors pas encore les questions de mise en page. Dès 1996 Internet Explorer 3 implémente partiellement ces spécifications. Netscape a suivi le mouvement en 1997 un peu contrainte si elle voulait survivre à la guerre des navigateurs qui se déroulait à cette époque entre les deux navigateurs.
- La spécification CSS 2 a suivi assez rapidement la précédente : elle est devenue une recommandation en mai 1998. Elle a étendu largement la portée du langage en rajoutant environ 70 propriétés, dont de grandes avancées telles que les propriétés de positionnement, mais aussi les types de média... Cette spécification a été un semi-échec car elle a pris beaucoup de temps pour être implémentée dans les navigateurs. Le W3C a donc revu sa copie avec la spécification CSS 2.1.
- L'échec de CSS 2 a conduit le W3C à corriger cette spécification avec CSS Level 2 Revision 1 (CSS 2.1). Le groupe de travail a pris cette fois plus de temps puisqu'il s'est lancé dans le travail en 2001 et la spécification CSS 2.1 est devenue une *Candidate Recommendation* en juillet 2007, statut qu'elle a conservé jusqu'à juin 2011 où elle est devenue une recommandation. Cette spécification a corrigé la précédente sur de nombreux petits points pour mieux coller à la pratique des navigateurs lors de l'implémentation de CSS 2. Mais cette spécification a pris beaucoup de temps pour être implémentée dans les navigateurs : par exemple Internet Explorer 7 l'implémente à 56 %, Firefox 3 à 93 % et Opera 9 à 94 % (15).
- Le travail sur CSS 3 a débuté dès 1999, et en parallèle à la révision de CSS 2.1. Le W3C a changé sa méthode de travail pour CSS 3 en rendant cette spécification modulaire : c'est-à-dire que cette spécification

n'est plus un tout géré de manière uniforme mais est traitée en modules indépendants les uns des autres. Toutes les règles communes à un domaine (comme les sélecteurs, les arrière-plans et bordures, les polices...) sont regroupées dans un module. Les modules sont indépendants les uns des autres, et donc ils se trouvent à différents stades d'avancement en fonction des priorités qui leur sont accordées par le groupe de travail du W3C.

Vous retrouverez l'état d'avancement de tous ces modules, leur contenu, ainsi que les priorités attribuées à chaque module sur cette page du W3C : [🇫🇷 CSS : état d'avancement du groupe de travail](#).

La plupart des modules sont encore à l'état de *Working Draft*, en conséquence ces spécifications sont encore peu implémentées dans les navigateurs.

Vous trouverez ici les principaux liens utiles sur le site de W3C :

- les spécifications concernant HTML et XHTML : [🇫🇷 la spécification HTML 4.01](#) et [🇫🇷 la spécification XHTML 1](#) ;
- pour tout savoir sur le travail du W3C concernant HTML : [🇫🇷 page d'accueil du W3C concernant HTML](#) ;
- les spécifications concernant les CSS 2.1 : [🇫🇷 http://www.w3.org/TR/REC-CSS2/](#)
- pour tout savoir sur le travail du W3C concernant les CSS : [🇫🇷 page d'accueil du W3C concernant les CSS](#) ;
- le validateur HTML : [🇫🇷 http://validator.w3.org/](#) ;
- le validateur CSS : [🇫🇷 http://jigsaw.w3.org/css-validator/](#).

## D. Avantages qu'apportent les feuilles de style à la création de sites web

Le CSS apporte de nombreux avantages, notamment :

- il permet de réduire la quantité de code, et donc de gagner en espace disque et en bande passante. La navigation se fait alors plus rapidement. Cela a aussi l'avantage de faciliter la maintenance ;
- la présentation est centralisée : il est donc plus facile de modifier rapidement le design, voire d'avoir plusieurs designs au choix de l'utilisateur ;
- le CSS permet d'adapter son site en fonction du type d'agent utilisateur qui visite le site ;
- la plupart des navigateurs permettent aussi d'influer de différentes manières sur le code CSS, permettant notamment d'améliorer l'accessibilité des sites web pour les malvoyants.

### 1. Réduction de la quantité du code

Le CSS réduit la quantité de code nécessaire.

En effet, à l'origine, la mise en forme se trouvait dans le code HTML grâce à certains éléments et attributs aujourd'hui obsolètes (exemple : balises `<font>` ou `<center>`, attributs `align`, `size`, `color`, `border`, `bgcolor`...). Ceux-ci devaient être répétés chaque fois qu'une mise en forme particulière était requise. Avec CSS, elle se trouve maintenant dans un fichier externe, s'applique à autant de pages (X)HTML que voulu, avec une règle CSS qui peut agir sur un nombre très important d'éléments dans la page avec un minimum d'interventions sur le code (X)HTML. Ce code (X)HTML se voit d'autant réduit de tous ces attributs de présentation.

De même l'utilisation de CSS dans la mise en page à l'aide de `<div>` allège considérablement le code (X)HTML en y supprimant l'usage des tableaux : outre le fait qu'il faut utiliser au minimum six balises pour faire un tableau ultraminimaliste, là où deux balises `<div>` suffisent, la présentation en tableaux de designs complexes nécessitait souvent l'utilisation d'un code lourd, avec de nombreux tableaux imbriqués.

Un tableau nécessite un minimum de six balises :

```
<table>
  <tr>
    <td>
Contenu
    </td>
```

Un tableau nécessite un minimum de six balises :

```
</tr>
</table>
```

Tous ces facteurs réunis font que le code (X)HTML est plus léger, on gagne en espace disque, et donc en bande passante.

La navigation est donc plus rapide, d'autant plus que les fichiers CSS sont téléchargés à la première visite d'un site puis mis en cache par le navigateur, qui va ensuite le récupérer du cache au lieu de le télécharger à chaque fois.

Vous pouvez le constater par vous-même avec **l'exemple 03 : Mise en page d'une structure XHTML avec l'ancienne méthode des tableaux** qui est une reprise de l'exemple numéro 2, mais en n'utilisant cette fois-ci que des attributs de présentation issus du langage HTML sans utilisation, à aucun moment, de CSS.

Pour faire un comparatif :

	Nombre de lignes	Nombre de caractères	Poids du fichier en octets
Code XHTML + CSS	126 lignes	5 330 caractères	5 580 octets
Code XHTML ancien	140 lignes	6 858 caractères	7 136 octets
<b>Économies faites</b>	10 %		22 %

 Bien sûr ces chiffres ont l'air relativement faibles, mais il faut les mettre en relation avec la relative simplicité du code pris en exemple. Sur une page web au design complexe les différences se feraient plus sentir.

## 2. Centralisation de la présentation

Un des grands intérêts du CSS est d'avoir l'ensemble de la présentation d'un site à un seul endroit. En effet le code CSS se retrouvera dans un ou plusieurs fichiers CSS, en fonction de l'organisation, donc séparé du code (X)HTML. Cela en facilite encore une fois la maintenance car si on veut changer un style on n'aura à le modifier que dans un fichier, et non dans l'ensemble des fichiers (X)HTML.

Cela permet aussi de changer le design complet d'un site en modifiant simplement le fichier CSS sans avoir à retoucher l'ensemble du code (X)HTML. Pour aller encore plus loin, en associant le CSS à un langage côté serveur, il est possible d'avoir plusieurs styles CSS pour un même site, grâce à un CSS style switcher, un excellent exemple en est le site CSS Zen Garden, qui montre également toute la puissance du CSS.

 Consulter le site [CSS Zen Garden](#) pour constater tout ce qu'il est possible de faire avec CSS, et aussi voir ce qu'est un CSS Style Switcher.

## 3. Adaptation du site en fonction de l'affichage

Le CSS permet d'organiser le design d'un site en fonction du périphérique qui va l'afficher. En spécifiant un type de média, vous pouvez proposer une feuille de style ou un code CSS optimisé pour un affichage sur un écran d'ordinateur (type de média screen), un autre optimisé pour l'impression de la page (type de média print), un autre optimisé pour les appareils disposant d'écrans de taille réduite (type de média handheld), etc.

## 4. Création de feuilles de style personnalisées

La plupart des navigateurs disposent dans leurs options d'un certain nombre de champs configurables qui vont influencer sur la présentation des pages web. Cela concerne principalement la taille et les types de police utilisés par défaut ainsi que les couleurs...

Certains navigateurs permettent également de créer et de mettre en œuvre une feuille de style personnalisée.

### Chemin vers les options :

- **Firefox** : menu Outils -> Option -> onglet Contenu.
- **Internet Explorer** : menu Outils -> Options Internet -> Onglet Général -> Section Apparences -> Boutons Couleurs, Police et Accessibilité.
- **Safari** : menu Modifier -> Préférences -> onglet Aspect.
- **Opera** : menu Outils -> Préférences -> Pages Web ainsi que Onglet avancé -> Contenu -> bouton Options de style.
- **Google Chrome** : bouton Personnaliser -> Options -> Options générales -> bouton "Modifier les paramètres relatifs à la police et à la langue".

Ces options permettent d'influer sur la feuille de style du navigateur et vont permettre d'adapter les sites à vos préférences dans le cas où l'auteur du site n'a pas déclaré de style spécifique pour ces éléments.

Cela peut même aller au-delà : par exemple Internet Explorer (toutes versions), Firefox et Opera permettent de mettre en œuvre votre propre feuille de style... Cela vous permet d'adapter les sites que vous consultez à vos préférences.

La mise en œuvre avec Internet Explorer est assez simple : allez dans les options, sur le bouton Accessibilité, il vous suffit de choisir le fichier css qui vous convient.

Dans Opera, il vous suffit de créer le fichier user.css, dans le dossier suivant (pour Windows) :

Chemin vers le fichier CSS personnalisable de Opera sur Windows :

```
C:\Program Files\Opera\styles\user.css
```

Ou de choisir votre propre fichier en l'indiquant dans les options.

Pour Firefox il faut accéder au dossier "*chrome*" se trouvant dans le profil de Firefox, que vous trouverez en règle générale à cet endroit (pour Windows) :

Chemin vers le fichier CSS personnalisable de Firefox sur Windows :

```
C:\Documents and Settings\Nom de l'utilisateur\Application Data\Mozilla\Firefox\Profiles\nom du profil\chrome\
```

Il faudra alors créer le fichier "userContent.css" et utiliser la syntaxe suivante :

Code à insérer pour sélectionner un site web :

```
@-moz-document url-prefix(http://www.developpez.net/forums/) {  
    /* Placez ici le code pour adapter la page */  
}
```

À la suite de url-prefix indiquez l'Uri vers le site ou vers le fichier que vous voulez adapter. Vous pouvez alors indiquer vos règles personnelles entre les accolades.

## E. Conventions et contenu de ce cours

Vous trouverez dans cette section les conventions mises en œuvre dans cette série d'articles sur CSS, notamment quant aux exemples.

## 1. Les exemples

J'illustre mes propos avec le maximum d'exemples. Cependant pour être le plus concis et le plus précis possible, ces exemples sont, pour plus de clarté, extrêmement succincts : seule la partie du code strictement nécessaire à la compréhension de l'exemple sera présente. Tous les autres éléments d'une page web, même minimalistes, comme l'entête, les autres éléments du corps... ne seront pas détaillés dans l'exemple.

En ce qui concerne les exemples de code CSS, lorsque les déclarations ne sont pas nécessaires à la compréhension de l'exemple elles seront remplacées par des termes génériques définissant par la même occasion leur signification : propriété : valeur. Comme dans l'exemple suivant :

Exemple de mise en œuvre des déclarations génériques :

```
div {
    propriété : valeur;
}
```

Par contre, la plupart des exemples cités dans le cours seront mis en relation avec un exemple externe, auquel vous pourrez accéder en suivant le lien, et dans lequel vous aurez accès au code de la page complète.

Ces exemples externes suivent ce modèle-type :

- 1 le titre de l'exemple ;
- 2 l'exemple lui-même avec sa concrétisation pratique ;
- 3 un encadré sous l'exemple regroupant deux choses :
  - a d'abord les explications nécessaires à la compréhension de l'exemple : ce qu'il faut savoir, retenir...
  - b et ensuite le code complet de l'exemple avec coloration syntaxique.

Cliquez sur l'image suivante pour la voir en grand :

**Exemple 01 : Décomposition d'une règle CSS : les sélecteurs**

**Le titre**

Ce paragraphe avec comme nom de classe 'paragraphe' fait l'objet d'une autre déclaration qui s'appliquera à tous les éléments <p> à qui on a attribué une classe avec comme valeur 'paragraphe'.

**L'exemple concret**

Ce paragraphe n'a pas de nom de classe, le style ci-dessus ne s'applique donc pas à lui.

**Les explications**

Dans une règle CSS, le sélecteur correspond à la partie de code en rouge ou décoloré ci-dessus. Il peut être d'origine ou personnalisé. Pour aller plus loin, voir les articles CSS/HTML associés une règle CSS ou s'appliquer.

Dans cet exemple, la propriété n'est pas utilisée sur les liens, des liens externes, le cadre visible en rouge ou gris à l'intérieur plus de tous les éléments <div>. Le sélecteur règle déclare une table de bord et la couleur rouge pour tous les paragraphes (autres 'p') possédant l'attribut class avec la valeur 'paragraphe'. Chez des éléments 'p' qui ne possèdent pas d'attribut class avec comme valeur 'paragraphe' ne seront appliqués le style par défaut et aussi celui-ci pour 'p' qui est visible.

**Le code source nécessaire pour reproduire l'exemple ci dessus**

```

1 <!-- Exemple 01 : Décomposition d'une règle CSS : les sélecteurs -->
2 <html>
3 <head>
4 <title> Exemple 01 : Décomposition d'une règle CSS : les sélecteurs </title>
5 </head>
6 <body>
7 <div style="display: inline-block; width: 100px; height: 100px; border: 1px solid red; background-color: #f0f0f0;">
8   Ceci est un exemple de contenu qui sera affecté par la règle CSS ci-dessus.
9 </div>
10 <p class="paragraphe"> Ce paragraphe avec comme nom de classe 'paragraphe' fait l'objet d'une autre déclaration qui s'appliquera à tous les éléments <p> à qui on a attribué une classe avec comme valeur 'paragraphe'.
11 <p> Ceci est un autre exemple de contenu qui ne sera pas affecté par la règle CSS ci-dessus. </p>
12 </body>
13 </html>

```

Copyright © 2011 Christophe F. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts. Cette page est déposée à la SACD.

Voici à quoi ressemble un modèle d'exemple

Tous les exemples sont testés avec les navigateurs suivants (système d'exploitation : Windows XP) :

- Internet Explorer depuis la version 6 et sauf la version 9 ;
- Firefox depuis la version 2 ;
- Opera depuis la version 8.5 ;
- Safari depuis la version 3 ;
- Google Chrome.

 Les exemples sont tous marqués comme valides XHTML et CSS. Il se peut cependant que ce ne soit pas le cas, essentiellement au motif que je souhaite vous montrer les effets de syntaxes qui ne sont pas valides. Ces exemples seront marqués d'un astérisque (\*) rouge présent à côté du logo du W3C pour lequel la validité n'est pas atteinte.

## 2. Mise en avant des noms d'éléments dans le cours

Pour plus de clarté, différents éléments composant les langages CSS et (X)HTML seront mis en exergue dans le corps des articles, en leur attribuant des couleurs bien distinctes :

- Les noms d'éléments (X)HTML comme `<body>` ou `<div>` seront représentés en **bleu**.
- Les noms d'attributs (X)HTML comme `class` ou `name` seront représentés en **rouge**.
- Les noms des propriétés CSS comme `font-weight` ou `list-style-type` seront représentés en **vert**.
- Enfin les valeurs du langage (X)HTML et CSS comme `'nomdeclasse'` ou `'12px'` seront représentés entre guillemets simples (') et en **noir**.

## 3. Liste des articles liés

Vous trouverez ci-dessous, dans le cadre bleu, la liste des articles de cette série.

## Remerciements

Le prochain article sera consacré à faire vos premiers pas dans l'univers du CSS : on verra les différentes syntaxes s'appliquant à une règle CSS. Accédez à cet article : [la syntaxe d'une règle CSS](#).

Tous mes remerciements vont à **ClaudeLELOUP** pour sa relecture et ses corrections, ainsi qu'à **Macmillenium**, **Bovino** et tous les autres membres de **l'équipe Web** que j'ai oubliés, pour leur soutien.

Commentez cet article :