

Introduction à XHTML

Il n'y aura pas de HTML 5. C'est ce qu'a confirmé le W3C (World Wide Web Consortium), l'organisme qui édite les recommandations des langages du Web : le HTML est mort en tant que tel. Certes, pendant des années, il a permis à tous de « bricoler » des pages web plus ou moins bien ficelées, mais il était devenu trop permissif, et surtout se caractérisait par un manque de rigueur assez flagrant. La rivalité entre Netscape et Microsoft a entraîné la création de balises propriétaires utilisables uniquement dans l'un des navigateurs, chacun s'ingéniant à créer le gadget qui lui attirerait le plus d'utilisateurs. Cette situation ne faisait que gêner les créateurs de pages web qui étaient obligés de prévoir des solutions alternatives aux balises manquantes en fonction du navigateur client. Il est évident que c'est la pression du e-commerce qui a provoqué la disparition de HTML, et que la naissance de XML (eXtensible Markup Language) l'a précipitée. Le XML aurait pu l'emporter tout de suite (car à terme c'est lui qui restera), mais c'était sans compter avec sa complexité et les problèmes de lecture qu'il pose aux utilisateurs avec des navigateurs d'anciennes générations. Il a donc fallu définir une alternative à tout XML.

Généalogie du XHTML

XHTML (eXtensible HyperText Markup Language) est un langage de balisage (dit aussi langage de marquage) qui permet de structurer le contenu des pages web dans différents éléments. Voilà une définition bien abstraite, reconnaissons-le, mais nous y reviendrons en détail dans la section suivante en présentant la notion de balisage.

Historiquement, les langages de balisage sont issus du langage SGML (Standard Generalized Markup Language) créé en 1986 pour structurer des contenus très divers.

Ce langage s'est révélé trop complexe pour être appliqué tel quel au Web, d'où la nécessité d'en créer une version allégée respectant les mêmes principes essentiels.

L'inventeur du HTML (1992), Tim Berners-Lee, l'avait conçu à l'origine comme un outil de structuration des contenus, principalement textuels, et non pas pour créer des présentations diversifiées. Ce sont les développements successifs, l'essor populaire du Web et la concurrence acharnée entre Netscape et Microsoft pour s'emparer du marché des navigateurs, qui ont détourné HTML de sa vocation première avec l'ajout d'éléments de design qui n'avaient rien à y faire. Voulant faire mieux que l'autre, chacun des deux grands a empilé des couches superflues sur HTML. Il est vrai que l'entrée du Web dans le grand public nécessitait de répondre à une demande d'interfaces graphiques plus esthétiques.

L'absence d'un langage particulier dédié uniquement à la présentation poussait effectivement les webmestres à utiliser tous les moyens pour créer des présentations visuelles agréables. L'apparition de CSS (Cascading Styles Sheets) en 1996 aurait dû résoudre le problème du détournement de HTML de sa destination première. Les mauvaises habitudes étaient prises et la facilité faisait le reste.

L'apparition de HTML 4, et particulièrement de sa version « strict » associée à l'emploi systématique de CSS 2 (publié en 1998), pouvait apporter une solution efficace à ce problème. La création de XML (eXtensible Markup Language) en 1998 et son succès dans de multiples domaines d'application ont conduit le W3C (World Wide Web Consortium) à créer le langage XHTML, non plus comme une nouvelle version de HTML, mais comme une reformulation de HTML en tant qu'application XML. Au niveau des éléments et des attributs disponibles, il existe à vrai dire très peu de différences entre HTML 4 strict et XHTML 1.1.

Les éléments, balises et attributs

Mais au juste comment fonctionne XHTML et qu'est-ce qu'un langage de balisage ?

Vous avez sûrement déjà utilisé un traitement de texte tel que Word. Votre texte peut comprendre des titres, des paragraphes, des images, des tableaux, et vous pouvez utiliser différentes polices de caractères et différentes tailles de caractères dans le même document. Le document final que vous avez réalisé ne laisse apparaître que le résultat de votre mise en page, mais en arrière-plan, votre traitement de texte a enregistré tous les paramètres de mise en page que vous avez utilisés en plus du texte lui-même.

Dans un langage de balisage, tout contenu, qu'il s'agisse de texte, d'image ou d'éléments multimédias les plus divers, doit être renfermé dans un élément. En XHTML, comme dans HTML, chaque élément a un nom déterminé et la liste des éléments utilisables est limitative et clairement définie dans la DTD (Document Type Definition) liée à la version utilisée du langage. C'est la grande différence entre XHTML et XML, langage dans

lequel c'est le programmeur qui crée ses propres éléments selon ses besoins. À quelques exceptions près, un élément a la structure suivante :

```
<nom_element> Contenu </nom_element>
```

Son contenu est précédé par une balise d'ouverture `<nom_element>` et suivi par une balise de fermeture `</nom_element>`. Toutes les balises d'ouverture (ou marqueur) commence par le signe `<` et se terminent par le signe `>`. La balise de fermeture fait de même mais le nom de l'élément est précédé d'un slash (`/`). Les navigateurs interprètent donc les contenus en fonction du nom de l'élément et attribuent un style par défaut à chacun de ses contenus.

Les caractéristiques de chaque élément peuvent être précisées par des informations complémentaires que l'on désigne en tant qu'attributs de l'élément. Il peut s'agir par exemple de la définition de la largeur, de la hauteur ou de l'alignement du contenu. Comme nous le verrons, un certain nombre d'attributs sont communs à quasiment tous les éléments de base dans les sections suivantes.

Les attributs d'un élément sont toujours définis dans la balise d'ouverture et doivent être séparés les uns des autres par au moins une espace typographique. Chaque attribut doit avoir une valeur, contrairement à ce qui se pratiquait dans HTML 4, même s'il ne peut prendre qu'une valeur unique. Aucune valeur n'est donc implicite du moment que l'attribut figure dans la balise d'ouverture. La présence de certains attributs est obligatoire dans quelques éléments particuliers, ce que nous préciserons systématiquement quand ce sera le cas. La plupart du temps, les attributs d'un élément sont facultatifs et c'est au programmeur de déterminer leur définition par rapport au cas qu'il doit traiter. Nombre d'attributs ont une valeur par défaut. Cela signifie que même si on ne les définit pas dans l'élément, celui-ci se comporte comme si nous avions défini explicitement cette valeur. Il est donc important de connaître ce type d'attribut et de ne pas négliger de les définir avec une autre valeur si ce comportement par défaut n'est pas désiré. La valeur de tous les attributs doit être définie entre des guillemets doubles. La syntaxe conforme d'un élément ayant des attributs est donc la suivante :

```
<nom_element attribut1="valeur1" attribut2="valeur2" > Contenu de l'élément  
  </nom_element>
```

Le contenu d'un élément peut être constitué de texte ou d'autres éléments qui, eux-mêmes, peuvent en contenir d'autres, et ainsi de suite. Cet ensemble d'inclusion constitue la hiérarchie du document XHTML.

Les attributs de base de XHTML

Dans leur quasi-totalité, les éléments disponibles en XHTML ont en commun un ensemble d'attributs ayant chacun le même rôle. Ces attributs se répartissent en trois catégories. Chaque élément peut avoir par ailleurs d'autres attributs particuliers. Quand nous définirons par la suite les différents éléments, nous signalerons s'ils possèdent ces attributs sans rappeler leur définition.

Les attributs courants (noyau)

Ils s'appliquent à quasiment tous les éléments. On compte trois principaux attributs courants et un quatrième encore accepté mais dont l'emploi est déconseillé :

- L'attribut `id`. Il sert à identifier un élément de manière unique en lui donnant un nom, soit pour lui attribuer un style, soit pour y faire référence sans ambiguïté dans un script JavaScript.
- L'attribut `class`. Il contient le nom d'une classe CSS qui contient des définitions de styles. Comme nous le verrons dans la deuxième partie, son usage est très répandu pour affecter des styles ponctuellement à un élément.
- L'attribut `title`. Il contient un texte qui apparaît dans une bulle quand l'utilisateur positionne le curseur quelques instants (ce n'est pas instantané) sur un élément. Le texte qu'il contient peut servir à fournir une information ou une explication sur le rôle de l'élément.
- L'attribut `style`. Il permet de définir un style localement pour un élément donné. Il est encore toléré en XHTML 1.1 mais déconseillé.

Les attributs d'internationalisation

- L'attribut `xml:lang`. Il qui remplace l'attribut `lang` de HTML 4.
- L'attribut `dir`. Il indique le sens de lecture du contenu textuel d'un élément ; il peut prendre les valeurs `ltr` (lecture de gauche à droite) ou `rtl` (de droite à gauche).

Les attributs de gestion d'événements

Ces attributs permettent de gérer les événements dont un élément peut être le siège et qui sont créés par l'utilisateur. Leur contenu est un script écrit en général en langage JavaScript. Les DTD HTML 4 définissent dix attributs de gestion d'événements, y compris pour des éléments qui ne peuvent pas être le siège de ces événements. Il appartient donc aux programmeurs d'effectuer des tests pour vérifier la réalité des événements pour un élément donné. Vous trouverez ci-après la liste des gestionnaires de base et la description de l'événement correspondant.

Tableau 1-1. Les attributs gestionnaires d'événements communs

attribut	Action de l'utilisateur
<code>onclick</code>	Clic sur le contenu de l'élément.
<code>ondblclick</code>	Double-clic sur le contenu de l'élément.
<code>onkeydown</code>	Maintien d'une touche enfoncée.
<code>onkeypress</code>	Frappe sur une touche.
<code>onkeyup</code>	Relâchement d'une touche enfoncée.

Tableau 1-1. Les attributs gestionnaires d'événements communs (suite)

attribut	Action de l'utilisateur
onmousedown	Enfoncement d'un bouton de la souris.
onmousemove	Le curseur de la souris bouge dans la zone de l'élément.
onmouseout	Le curseur de la souris quitte la zone de l'élément.
onmouseover	Le curseur de la souris est au-dessus de la zone de l'élément.
onmouseup	Relâchement d'un bouton de la souris au-dessus de la zone de l'élément.

D'autres attributs gestionnaires d'événements sont spécifiques à certains éléments, que nous citerons au fur et à mesure de notre étude de ceux-ci.

Intérêt des spécifications

La création du XHTML ne consiste pas seulement en la redéfinition de quelques règles syntaxiques pour aménager le langage HTML. S'il ne s'agissait que de cela, cet ouvrage n'aurait pas lieu d'être. Il s'agit bien plus selon moi d'un changement de pensée et d'organisation qui doit s'opérer dans la création des pages web.

Une page web créée avec XHTML doit être pensée en distinguant deux parties :

- Un contenu, structuré au moyen des éléments XHTML (grandes divisions, titres, paragraphes, tableaux, images et liens, etc.). À ce stade, et même s'il en a déjà une idée, le créateur ne doit pas nécessairement avoir une idée définitive de la présentation finale. Il lui faut maîtriser principalement l'organisation des informations à fournir à un utilisateur.
- Une feuille de style CSS, définissant la mise en page de ces éléments en fonction du média qui va opérer le rendu du contenu (polices et tailles de caractères, bordures, marges, couleurs, positionnement dans la page, etc.). Les médias se diversifient en effet de plus en plus en devenant des éléments portables dotés de petits écrans, et il n'est pas impossible que le traditionnel écran d'ordinateur ne soit plus à l'avenir le principal vecteur d'affichage d'une page web. La séparation du contenu et de la présentation étant réalisée, il est possible d'associer à chaque média une feuille de style adaptée au terminal.

L'utilisation de ces méthodes présente les avantages suivants :

- Une meilleure organisation du contenu.
- Une meilleure qualité du code et plus grande rapidité d'affichage sur les navigateurs récents (Firefox, Mozilla, Internet Explorer, Netscape...).
- Une réduction des coûts de développement et de maintenance des sites web ainsi qu'une réutilisabilité accrue et rapide du code. En effet, en ayant respecté les principes

précédents, il est très facile de modifier rapidement toute la présentation d'une page sans toucher au code XHTML.

Les standards XHTML et CSS sont aujourd'hui incontournables pour tous ceux qui veulent concevoir un site web de manière professionnelle, et tous les étudiants en informatique et les professionnels du Web se doivent d'acquérir ou de mettre à jour leurs connaissances sur ces techniques.

Règles de base XHTML

Un document bien formé

Un document XHTML doit respecter certaines règles simples :

- Les éléments et les attributs sont sensibles à la casse et doivent être écrits en minuscules. Par exemple, `<body>` et non plus `<BODY>` comme en HTML.
- Les éléments non vides doivent avoir obligatoirement une balise d'ouverture et une balise de fermeture. Par exemple, on ne doit plus écrire :

```
<ol>
  <li>Item 1
  <li>Item 2
```

mais le code suivant :

```
<ol>
  <li>Item1 </li>
  <li>Item2 </li>
</ol>
```

- Les éléments vides ne comportent qu'une seule balise et doivent se terminer par les caractères `</>` précédés d'une espace pour marquer la fin de l'élément. Par exemple, il ne faut plus écrire :

```
<img src= "monimage.gif"> <hr> <br>
```

mais le code suivant :

```
<img src= "monimage.gif" /> <hr /> <br />
```

- Les éléments ne doivent pas se chevaucher. Ils doivent donc obéir au principe premier-ouvert-dernier-fermé. Dans ce cas, le premier élément est le parent du second et celui-ci est enfant du premier. Par exemple, le code suivant est incorrect :

```
<div> Cette division contient un titre <h1> Important ! </div> </h1>
```

et doit être remplacé par :

```
<div> Cette division contient un titre <h1> Important ! </h1></div>
```

- Tous les attributs doivent avoir une valeur incluse entre des guillemets doubles ("). Les différents attributs du même élément doivent être séparés par au moins une espace. Par exemple, il ne faut plus écrire :

```
<p class=styleperso title=attention> Texte important</p>
```

mais le code suivant :

```
<p class="styleperso" title="attention" > Texte important</p>
```

- À tous les attributs utilisés doit être donnée une valeur, y compris ceux dont la valeur est unique. Par exemple, il ne faut plus écrire :

```
<input type= "checkbox" checked disabled />
```

mais le code suivant :

```
<input type= "checkbox" checked="checked" disabled="disabled" />
```

- L'attribut `name` qui servaient à identifier certains éléments (`<a>`, `<form>`, par exemple) est supprimé et doit être remplacé par l'attribut `id`.
- Les scripts et les feuilles de style qui contiennent les caractères `<` et `&` peuvent figurer dans des sections CDATA de la façon suivante :

```
<script type="text/javascript">  
  <![CDATA[  
    Code du script...  
  ]]>  
</script>
```

Ces sections n'étant pas actuellement reconnues par les navigateurs, nous ne les utiliserons pas dans cet ouvrage, mais il faudra en tenir compte dans un avenir proche. Une autre solution efficace consiste à inclure les scripts ou les feuilles de style contenant ces caractères dans des fichiers séparés et à les inclure à l'aide de l'élément `<link>` sur lequel nous reviendrons en détail par la suite.

Un document conforme

Un document XHTML se doit également de respecter les règles d'inclusion des éléments les uns dans les autres, telles qu'elles sont définies dans la DTD choisie. En effet, une DTD définit la liste limitative de tous les éléments XHTML utilisables et énumère ceux qui peuvent y être inclus. Le respect de ces contraintes est impératif pour que le document soit déclaré conforme à la DTD. Vous trouverez à cet effet tout au long de cet ouvrage lors de la description des éléments, et dans l'annexe A, non pas le texte des DTD XHTML 1 qui est particulièrement difficile à lire, mais son interprétation, pour chaque élément, sous la forme de la liste de ses éléments enfants (ceux qu'il peut inclure) et de ses éléments parents (ceux dans lesquels il peut être inclus).

Validation d'un document : le label du W3C

Malgré toutes les vérifications auxquelles vous pouvez procéder personnellement, il peut rester des erreurs de conformité dans votre code. Comme peut le faire un compilateur qui signale les éventuelles erreurs de syntaxe, le validateur du W3C permet de vérifier si le code est bien formé et conforme à la DTD indiquée. Pour lancer cette validation automatique, vous devez soumettre l'URL ou le code de vos documents XHTML au validateur du W3C accessible à l'adresse suivante : <http://validator.w3.org>

La figure 1-1 montre la page de validation du site du W3C dans laquelle plusieurs moyens sont mis à disposition pour valider un document. Vous pouvez saisir l'URL de la page si le document est déjà transféré sur un serveur (repère ❶), choisir le fichier sur votre ordinateur en cliquant sur le bouton Parcourir (repère ❷), ou encore écrire le code à vérifier directement dans une zone de saisie (repère ❸).

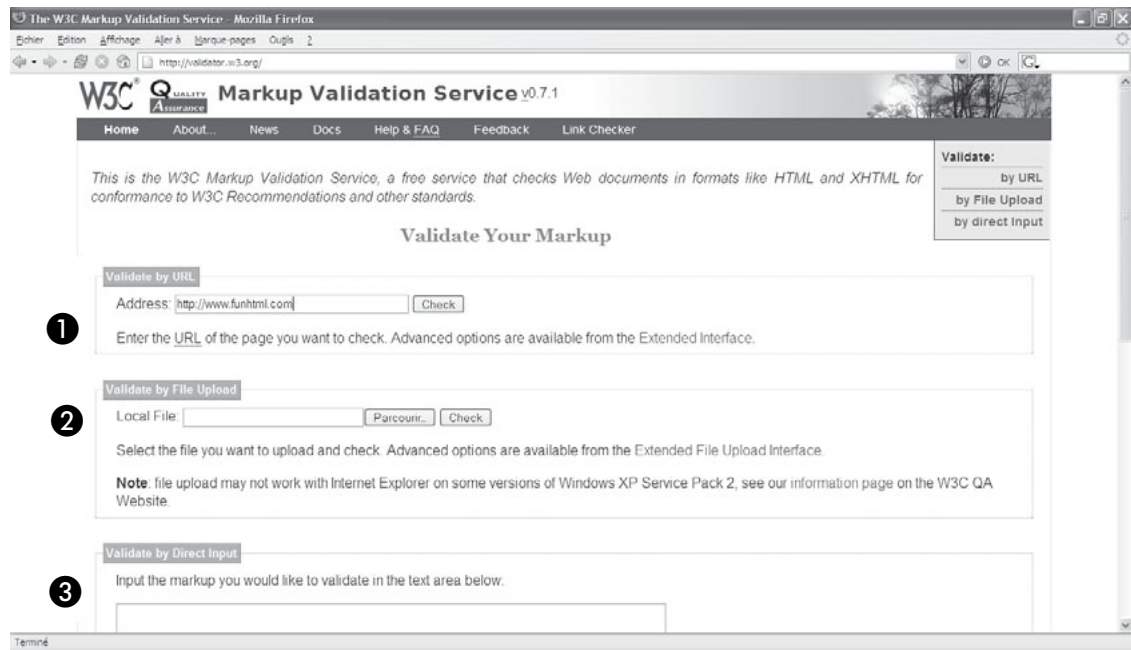
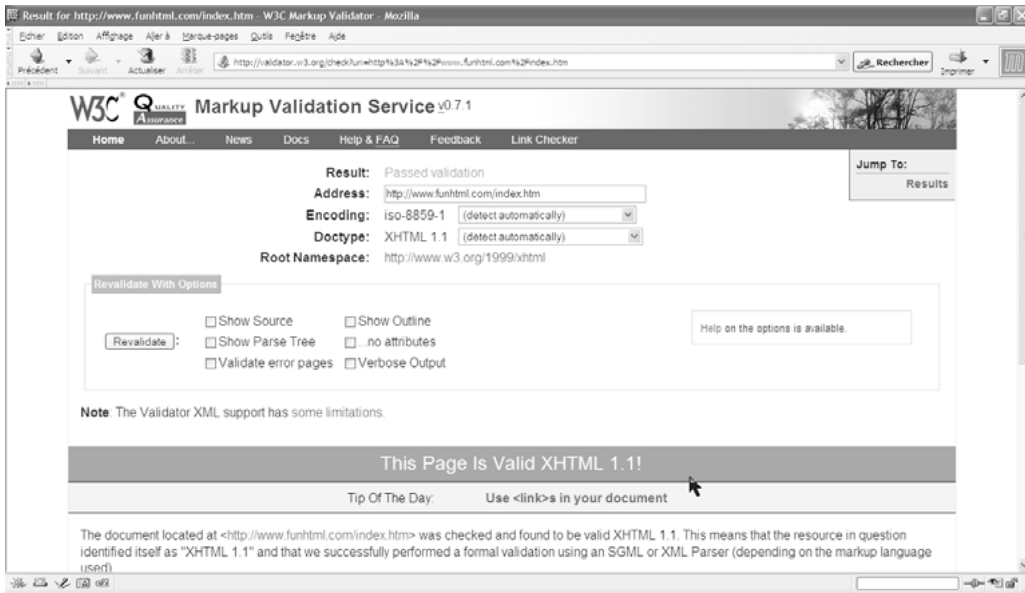


Figure 1-1

La page de validation du W3C

Le résultat de la validation est affiché dans une nouvelle page ; si le code est conforme, vous obtenez une page semblable à celle de la figure 1-2, sinon la liste des erreurs ainsi qu'un commentaire sont fournis.

**Figure 1-2**

La page de résultat de validation du W3C

L'environnement de travail

Pour créer des pages web avec XHTML et CSS, il faut être doté d'un environnement de travail adapté. En théorie, un simple éditeur de texte tel que le Bloc Notes de Windows ou Emacs sous Linux par exemple peut suffire. Cependant, comme la saisie des noms de différents éléments XHTML dans ce type d'éditeur peut devenir à la longue plutôt rébarbative, nous mentionnons quelques outils susceptibles de faire gagner du temps.

Les éditeurs visuels

Dans ce type d'éditeurs, vous travaillez graphiquement sur une page en y incluant des éléments sans taper une ligne de code. Le plus connu est Dreamweaver, mais il en existe bien d'autres plus ou moins perfectionnés tel que Golive. L'inconvénient de ces éditeurs tient en fait à ce qui paraît être leur avantage : le code que l'on ne saisit pas est généré automatiquement et rien ne garantit qu'il convienne ou qu'il soit conforme aux dernières spécifications XHTML, car même Dreamweaver permet l'inclusion d'éléments obsolètes. De plus, toute génération automatique éloigne du travail de programmation et ne présente pas un avantage évident pour la création de pages, alors que ce peut être le cas dans un éditeur dédié à un langage de programmation, Java par exemple, et qui peut décharger le programmeur de tâches répétitives.

Outre le temps d'apprentissage et son prix élevé, je ne pense pas qu'il soit utile de faire appel à ce genre d'éditeur qui aurait tendance à vous rendre passif.

Signalons également dans cette catégorie, l'existence d'un nouvel éditeur qui se veut le concurrent libre et gratuit de Dreamweaver pour la création de pages web. Il s'agit de Nvu, téléchargeable sur le site <http://www.nvu.com>. Dans sa version actuelle, version 1.0, il est encore assez loin de son modèle mais ses débuts sont prometteurs car il offre des fonctionnalités avancées, comme un éditeur de styles CSS. C'est d'ores et déjà un bon produit dont il faudra suivre les évolutions futures. La figure 1-3 présente l'espace de travail de Nvu.

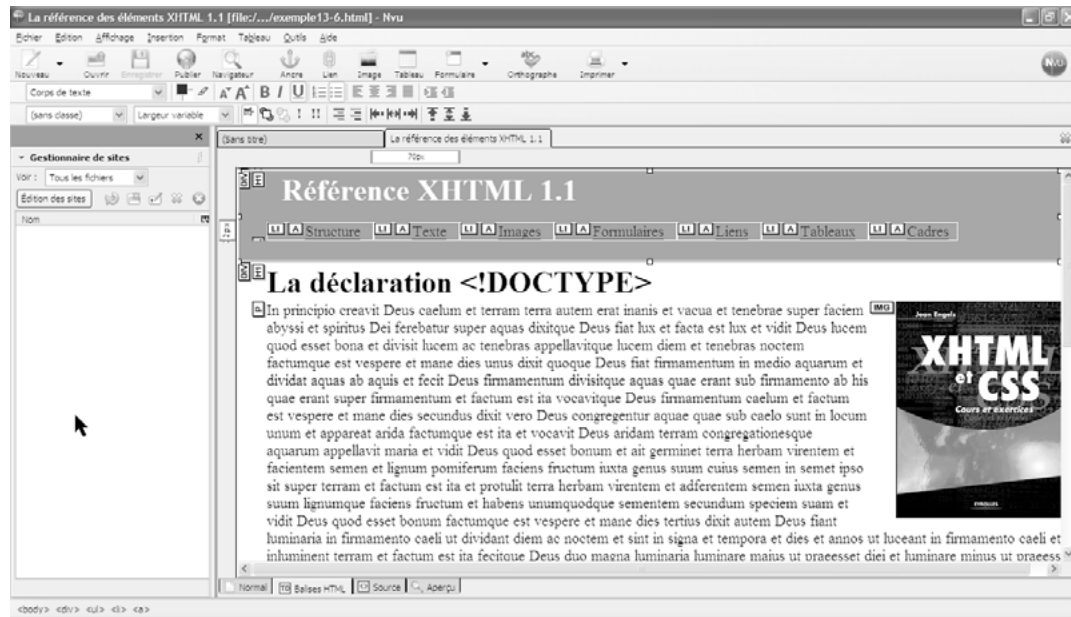


Figure 1-3
L'éditeur Nvu 1.0

Les éditeurs classiques

Entre le Bloc Notes Windows et Dreamweaver, certains éditeurs offrent un compromis, apportant à la fois une aide à la saisie en dispensant de taper soi-même le nom des éléments et en laissant la liberté du choix de ces éléments. Outre le fait qu'ils présentent souvent l'avantage d'être gratuits, ces éditeurs, très nombreux et disponibles en téléchargement, permettent de créer rapidement la structure d'un document en utilisant un squelette commun à toutes les pages XHTML. Nous donnerons la structure de base d'une telle page au chapitre 2. Il vous restera ensuite à inclure dans le corps de la page les différents éléments qui vont structurer son contenu. Si vous avez procédé à une analyse préalable sur le papier, comme il se doit avant tout codage, cette phase de travail d'inclusion sera très rapide. Parmi les nombreux éditeurs, j'ai choisi pour ma part EditPlus qui s'avère très pratique pour incorporer rapidement les différents éléments XHTML et CSS. La figure 1-4 montre l'aspect de l'environnement de travail qu'il fournit. La fenêtre de

l'éditeur est divisée en plusieurs zones. La zone ❶ contient le code de la page. Comportant plusieurs onglets, la zone ❷ permet de visualiser selon vos besoins du moment la liste des éléments XHTML 1.1, celle des propriétés CSS 2 ou un explorateur de fichiers. La zone ❸ contient en plus des menus habituels un ensemble de boutons permettant l'inclusion instantanée du code correspondant à une icône (lien, titres, tableaux, etc.). Pour incorporer un élément XHTML complet, muni donc d'une balise d'ouverture, une balise de fermeture éventuelle ainsi que certains attributs importants, il vous suffit de placer le curseur à l'endroit désiré dans le code, puis d'effectuer un double-clic sur l'élément dans la zone de gauche (repère ❷). Il en est de même pour créer une feuille de style : suffit d'un clic sur la propriété CSS voulue. La liste des éléments XHTML et CSS fournie par défaut avec EditPlus est actuellement plus large que celle des éléments conformes de XHTML 1.1, mais vous pouvez la personnaliser en enlevant des éléments ou en en ajoutant. Si vous utilisez par exemple très souvent tel attribut d'un élément, vous pouvez le rajouter, et il apparaîtra ensuite systématiquement quand vous choisirez cet élément, car la liste des éléments et de leurs attributs est enregistrée dans un fichier. Pour effectuer cette opération, effectuez un clic droit sur le nom d'un élément et choisissez Edit. Vous pouvez ensuite procéder à toutes les modifications désirées. Vous pourrez télécharger sur le site funxhtml.com/edit les fichiers `xhtml11.ctl` et `css21.ctl` et suivre la procédure indiquée sur le site pour obtenir dans EditPlus tous les éléments XHTML 1.1 et les propriétés CSS 2.1 qui sont actuellement conformes aux recommandations du W3C.

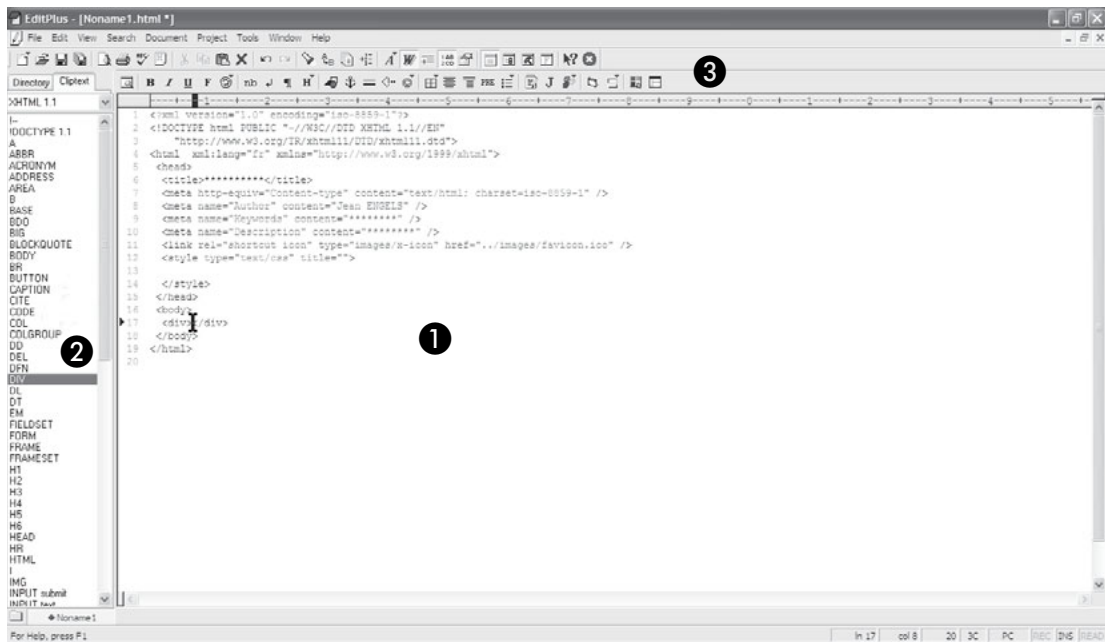


Figure 1-4

L'éditeur EditPlus

Tests et mise en place du site

Une fois que vous aurez créé l'ensemble des pages de votre site, il faudra vous assurer qu'elles ont un aspect identique ou du moins très semblable dans les différents navigateurs du marché et les résolutions d'écran afférentes. Ce dernier point doit d'ailleurs avoir fait l'objet d'un choix initial, en particulier si les dimensions choisies pour les différents éléments de la page sont définies de manière fixe (en pixels par exemple), et non en pourcentages, ce qui représente la meilleure solution. Choisir de créer des mises en page en ciblant le format le plus répandu (1024 par 768 pixels actuellement par exemple) peut produire des résultats désagréables si le poste client est en 800 par 600 pixels ou moins encore.

Si cette phase de test vous amène à constater des résultats divergents et indésirables, c'est sans doute que vous avez utilisés des fonctionnalités non encore prises en compte par un navigateur particulier, et le plus souvent il s'agira de propriétés CSS encore marginales dans Internet Explorer. Il vous appartiendra alors de renoncer à certaines d'entre elles au moins provisoirement, pour assurer l'universalité de vos pages si tel est votre objectif. Nous signalons à cet effet dans la deuxième partie de cet ouvrage les propriétés qui risquent de poser problème dans certains navigateurs.

Quand cette phase de test a été effectuée, il ne reste plus qu'à transférer l'ensemble des pages du site vers le serveur distant qui va les héberger pour les mettre à la disposition de tous. Vous devez pour cela utiliser un logiciel de transfert FTP (File Transfert Protocol).

Les hébergements gratuits

Les hébergeurs gratuits de sites personnels peuvent mettre à votre disposition des pages spécialisées permettant ce transfert via une interface web simple. Si vous disposez d'un nom de domaine sur un serveur dédié ou mutualisé, il vous faut généralement utiliser un logiciel FTP.

Il existe de nombreux logiciels de ce type, gratuits ou payants. Je recommande pour ma part d'utiliser FileZilla téléchargeable sur le site <http://filezilla.sourceforge.net/>, gratuit et très pratique. La figure 1-5 présente l'interface de FileZilla. Pour établir une connexion avec le serveur et acquérir les droits en vue d'effectuer les transferts de vos fichiers XHTML, de vos images ou des supports multimédias, vous devez connaître les paramètres de connexion qui sont fournis par l'hébergeur du site. Il s'agit des données suivantes :

- L'adresse du serveur ftp, par exemple `ftp.funhtml.com` (repère ❶).
- Le nom d'utilisateur, par exemple `funhtml` (repère ❷).
- Le mot de passe d'accès au site (repère ❸).

- Éventuellement, indiquez le port de communication (repère ④) (en général, il s'agit de la valeur 21).

Si ces paramètres sont correctement reconnus, vous avez accès à un gestionnaire de fichiers côté serveur similaire à celui que vous pouvez connaître dans Windows ou Linux (repère ⑤). Dans la partie gauche, vous trouvez le même type de gestionnaire de fichiers pour sélectionner les fichiers présents sur votre ordinateur (repère ⑥). Il vous suffit alors de réaliser une opération de glisser-déposer entre ces deux zones pour que le transfert commence. Sa durée dépend évidemment de la taille du fichier et de la vitesse de votre connexion. Le dossier principal côté serveur est nommé `www`. C'est dans ce dossier que le serveur ira chercher le fichier nommé `index.html` ou `index.htm` en réponse à la requête générale effectuée sur votre site sous la forme `http://www.votresite.com`. C'est dans ce fichier que vous devez créer la page d'accueil du site (ou la page `index.php` si elle contient du code PHP). Dans ce dossier `www`, vous pouvez créer autant de sous-dossiers que vous le désirez. Si vous constituez par exemple un sous-dossier nommé `xhtml`, son contenu principal devra figurer dans un nouveau fichier nommé encore `index.html`, et il sera alors accessible directement par la requête `http://www.votresite.com/xhtml`. Dans les deux cas (répertoire principal ou sous-répertoire), si vous ne créez pas de fichier nommé `index.html`, l'utilisateur devra écrire une requête complète comprenant le nom du fichier auquel il veut accéder de la forme `http://www.votresite.com/xhtml/page1.html`.

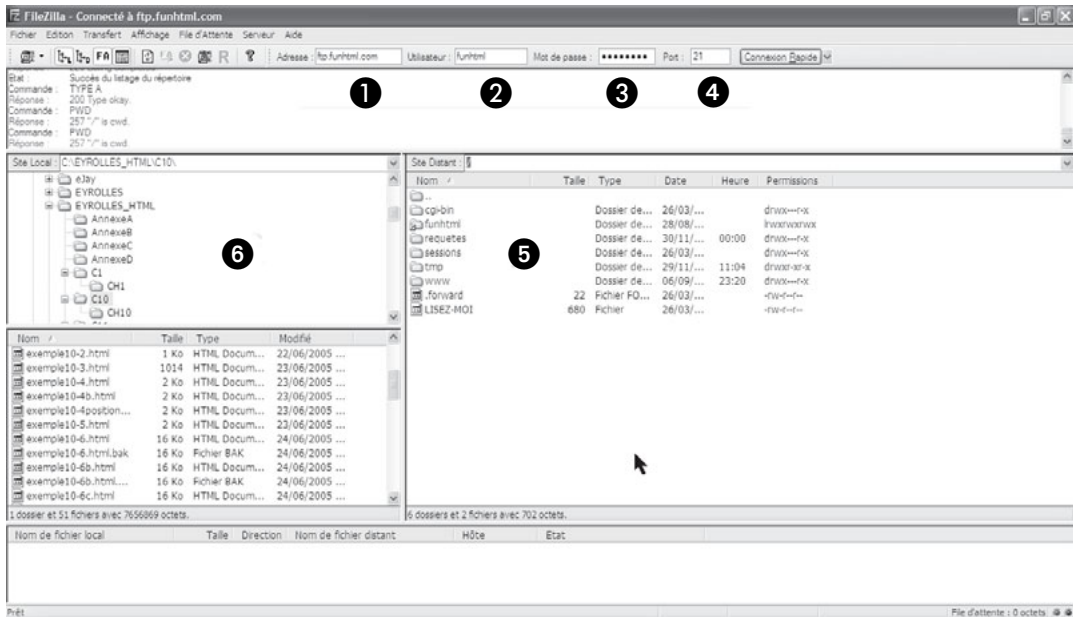


Figure 1-5
L'interface de FileZilla

Référencement du site

Une fois que vous avez créé votre superbe site conforme à XHTML et CSS, il reste encore une phase des plus importantes à accomplir, à savoir le faire connaître. À moins de disposer de moyens publicitaires conséquents, la fréquentation souhaitée ne se produira pas du jour au lendemain. Même si c'est l'adaptation du contenu à un besoin qui fait venir et revenir les internautes, il vous faut d'abord faire connaître l'existence de votre site.

Le moyen le plus simple, et généralement gratuit, est de le référencer dans les annuaires et moteurs de recherche les plus connus comme Google ou Yahoo afin qu'il soit bien placé en réponse à la recherche d'un internaute.

Des ouvrages entiers sont consacrés au référencement, apportant des conseils avisés pour augmenter la fréquentation d'un site. Je recommande le site <http://www.abondance.com> ou de se procurer l'ouvrage de son webmestre portant sur le référencement. Nous retiendrons cependant les quelques points suivants :

- Si vous achetez un nom de domaine – ce qui, lié à un hébergement mutualisé, est aujourd'hui possible à bon marché et « présente mieux » que l'hébergement gratuit offert par votre fournisseur d'accès dont les adresses sont très longues –, choisissez de préférence un nom court et facile à mémoriser. Les noms longs, et particulièrement s'ils sont composés de plusieurs mots, posent soucis à l'utilisateur qui se demande alors si les mots se suivent ou sont séparés par des tirets, d'où des interprétations divergentes et des erreurs. Certains moteurs de recherche affichent les sites répondant aux mêmes critères par ordre alphabétique et il vaut mieux que votre nom de domaine commence par « a » plutôt que par « z ».
- Les extensions .com (ou .fr en France) sont préférables à .org ou .net car ce sont les premières qui viennent à l'esprit des internautes s'ils ont oublié l'extension réelle. La notion de site commercial qui était liée à l'extension .com a désormais disparue et votre site ne peut avoir aucun aspect commercial et posséder cette extension.
- La définition d'un maximum de mots-clés rapportant objectivement le contenu de votre site est essentielle. Elle doit être réalisée à l'aide de l'élément :

```
<meta name="keywords" content="liste des mots clés" />
```

(voir le chapitre 2), qui est situé dans l'en-tête du document. Cette liste mérite toute votre attention car elle est utilisée par les moteurs de recherche pour indexer vos pages et mettre en adéquation la demande d'un internaute avec les sites qui lui correspondent. Les mots-clés se doivent de refléter fidèlement le contenu du site et les idées qui lui sont associées, et rien d'autre. Inutile par exemple de rechercher les mots les plus recherchés et de les inclure dans votre liste pour attirer du monde. Outre que vous risquez de tromper les internautes, vous pouvez surtout les décevoir si votre contenu ne correspond pas à leur attente. En revanche, n'hésitez pas à fournir une longue liste de mots-clés avec leurs variantes masculin/féminin et singulier/pluriel, ainsi que les mots dérivés qui peuvent correspondre à votre contenu. Vous augmenterez ainsi vos chances de figurer en bonne place dans les résultats des recherches effectuées par les internautes.

Les moteurs de recherche se basant aussi sur le contenu des pages, rien ne vous empêche de faire apparaître plusieurs fois dans le contenu les mots importants. Une astuce consistait à répéter ces mots dans la page et de les écrire de la même couleur que le fond de la page en créant des styles CSS appropriés et non plus les balises et les attributs utilisés pour définir les couleurs. Leur répétition est ainsi invisible dans la page, mais perçue par les robots qui analysent le texte.

- Créez la structure de vos pages en évitant l'emploi des cadres, autrefois très en vogue, mais dont les contenus sont mal indexés par les moteurs de recherche. L'emploi de XHTML et du positionnement CSS rend ces techniques caduques et des solutions de remplacement peuvent être mises à profit (voir le chapitre 13).
- Référez votre site dans tous les principaux moteurs de recherche. La figure 1-6 montre par exemple la page de référencement de Yahoo qui permet de soumettre un site gratuitement.

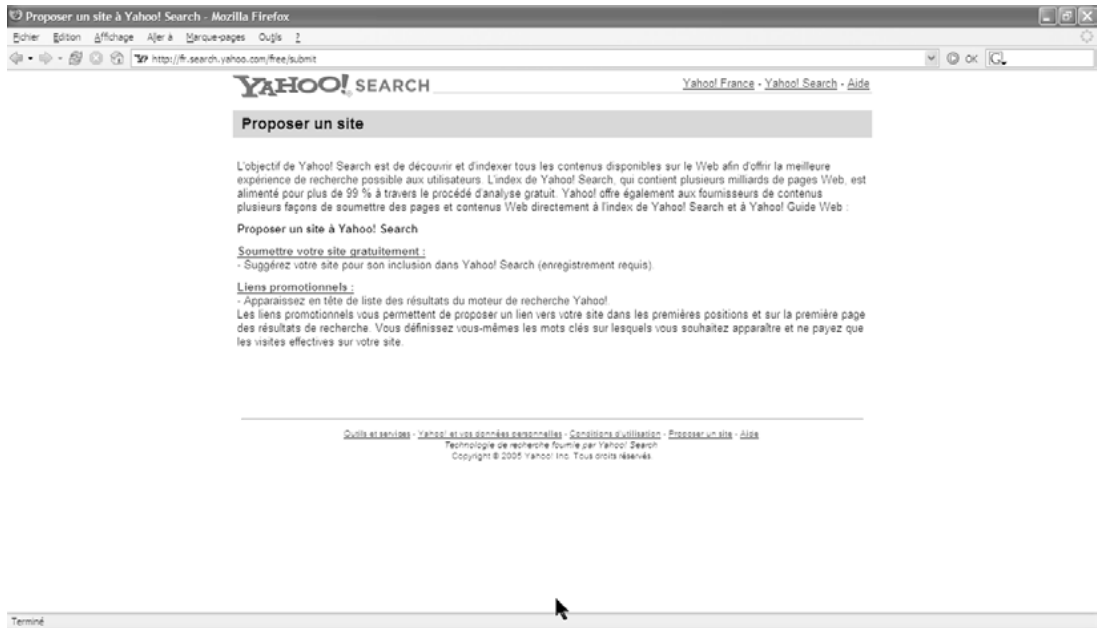


Figure 1-6

La page de référencement de Yahoo