La lecture de cette introduction pourrait ne pas paraître à certains comme totalement indispensable à la compréhension du reste de l'ouvrage. Pourtant, connaître la genèse et l'évolution de HTML et des langages apparentés peut vous simplifier grandement la vie, en vous permettant de mieux comprendre les préoccupations et buts sous-jacents des créateurs et développeurs du langage. Pour la création d'un site Web comme pour n'importe quelle activité de développement, connaître l'historique et la philosophie du langage employé procure un avantage souvent déterminant pour une réussite parfaite.

1.1. Bref historique de HTML

Le langage HTML (*HyperText Markup Language*) a été développé initialement par Tim Berners-Lee, alors au CERN. Il a rapidement connu un vif succès grâce au navigateur Mosaic, développé au NCSA. Pendant les années 1990, il a poursuivi sa croissance en profitant de celle, explosive, du Web, et s'est enrichi de nombreuses manières. Le Web repose sur le respect, par les concepteurs de pages et les éditeurs de logiciels, de conventions identiques pour HTML, ce qui a motivé le travail commun sur les spécifications de HTML.

C'est un format non propriétaire fondé sur SGML (*Standard Generalized Markup Language*) se conformant au standard international ISO 8879. Il peut être créé et traité par de nombreux outils, depuis des éditeurs de texte simples jusqu'à des outils dédiés sophistiqués WYSIWYG (*What You See Is What You Get* ou tel écran, tel écrit). HTML emploie des balises (comme `<H1>` et `</H1>`) pour structurer un texte en en-têtes, paragraphes, listes, liens hypertextes, etc.

La spécification HTML 2.0 (RFC 1866 de novembre 1995) a vu le jour sous le contrôle de l'IETF (*Internet Engineering Task Force*). Le groupe de travail HTML du W3C (*World Wide Web Consortium*) diffuse en janvier 1997 la spécification HTML 3.2, qui apporte plusieurs améliorations et modifications.

États d'avancement d'un document W3C

Tout document W3C passe par plusieurs états d'avancement (ou de maturité) portant un nom précis.

- **Brouillon** (WD, *Working Draft*) : un document publié par le W3C afin qu'il soit examiné par la communauté composée des membres du W3C, du public et des autres organismes techniques.
- **Candidat à la recommandation** (CR, *Candidate Recommendation*) : le W3C considère que le document a été largement commenté et répond aux exigences techniques du groupe de travail (*Working Group*). Il publie un CR pour obtenir des commentaires sur les mises en œuvre.
- **Proposition de recommandation** (PR, *Proposed Recommendation*) : c'est désormais un document technique mature, dont la possibilité de mise en œuvre a été largement vérifiée. Il est envoyé au Comité Consultatif (*Advisory Committee*) pour adoption finale.

- **Recommandation** (REC, *Recommendation*) : une Recommandation W3C est une spécification ou un ensemble de règles ayant reçu l'approbation du W3C. W3C en recommande un large déploiement. Elle est analogue à un standard tel que diffusé par d'autres organismes.

Viendra ensuite la spécification HTML 4, un progrès immense par rapport aux versions antérieures. Ses principales innovations concernent l'internationalisation, l'accessibilité, les tableaux, les documents composés, les feuilles de style, les scripts et l'impression.

- **Internationalisation** : les documents peuvent être écrits en toutes les langues et diffusés partout dans le monde. Cela a été accompli en tenant compte du document RFC 2070, qui traite de l'internationalisation de HTML. L'adoption de la norme ISO/IEC:10646 comme jeu de caractères du document pour HTML a représenté une étape importante. C'est la norme mondiale la plus complète, qui tient compte des problèmes concernant la représentation des caractères internationaux, le sens des écritures, la ponctuation et autres particularités des langues mondiales. Cela permet une indexation des documents par les moteurs de recherche, une typographie de qualité, la synthèse de la parole à partir du texte, la césure, etc.
- **Accessibilité** : au fur et à mesure de la croissance de la communauté du Web et de la diversification des capacités et aptitudes de ses membres, il devient crucial que les technologies employées soient appropriées à leurs besoins spécifiques. Le langage HTML a été conçu pour rendre les pages Web plus accessibles à ceux qui présentent des déficiences physiques. Les développements de HTML 4 qui ont été inspirés par le souci de l'accessibilité comprennent :
  - une meilleure distinction de la structure et de la présentation du document, en encourageant pour cela l'utilisation des feuilles de style au lieu des éléments et attributs de présentation HTML ;
  - l'amélioration des formulaires, ce qui comprend l'ajout de touches d'accès rapide (*access keys*), la possibilité de regrouper sémantiquement les contrôles des formulaires et les options de l'élément `SELECT` ainsi que l'ajout des étiquettes actives (*active labels*) ;
  - la possibilité de baliser la description textuelle d'un objet incorporé (avec l'élément `OBJECT`) ;
  - un nouveau mécanisme d'images cliquables côté client (l'élément `MAP`), qui permet aux auteurs d'intégrer des liens sous forme de texte et d'images ;
  - l'accompagnement obligatoire des images incluses avec l'élément `IMG` et des images cliquables incluses avec l'élément `AREA`, d'un texte de remplacement, ainsi que des descriptions longues des tableaux, images, cadres, etc. ;

— la gestion des attributs `title` et `lang` pour tous les éléments, ainsi que des éléments `ABBR` et `ACRONYM` ;

— un éventail élargi des médias cibles (tty, braille, etc.) à utiliser avec les feuilles de style ;

— l'amélioration des tableaux, en y incluant des légendes, des regroupements de colonnes et des mécanismes pour faciliter la restitution non visuelle ;

- **Tableaux** : le nouveau modèle de tableau est fondé sur le document RFC1942. Les auteurs disposent maintenant d'un plus grand contrôle sur leur structure et leur disposition (par exemple, les regroupements de colonnes). Ils peuvent spécifier la largeur des colonnes et permettre aux agents utilisateurs d'afficher les données de table progressivement, au fur et à mesure du chargement, plutôt que d'attendre le chargement entier du tableau.
- **Documents composés** : le langage HTML offre maintenant une structure standard pour l'incorporation d'objets média et d'applications génériques dans les documents HTML. L'élément `OBJECT` (de même que ses ancêtres, les éléments plus spécifiques `IMG` et `APPLET`) fournit le moyen d'inclure des images, des séquences vidéo, de l'audio, des mathématiques, des applications spécialisées et d'autres objets dans un document. Il permet aussi aux auteurs de spécifier une hiérarchie de restitutions de remplacement aux agents utilisateurs qui ne gèrent pas certaines restitutions particulières.
- **Feuilles de style** : les feuilles de style simplifient le balisage HTML et soulagent grandement HTML des responsabilités de la présentation. Elles donnent aux auteurs comme aux utilisateurs le contrôle de la présentation des documents (informations sur les polices de caractères, alignement, couleurs, etc.). Les informations de styles peuvent être spécifiées pour un élément ponctuel ou pour des groupes d'éléments. Elles peuvent se trouver à l'intérieur du document HTML ou dans une feuille de style externe. Les mécanismes qui associent une feuille de style à un document sont indépendants du langage de feuille de style. Avant l'apparition des feuilles de style, les auteurs disposaient d'un contrôle limité sur la restitution des pages. HTML 3.2 comprenait un certain nombre d'attributs et d'éléments permettant un contrôle de l'alignement, de la taille de la police de caractères et de la couleur du texte. Les auteurs abusaient également de tables et d'images pour la mise en pages. Le temps relativement long nécessaire pour que les utilisateurs mettent à jour leurs navigateurs implique que ces usages vont perdurer encore pendant un certain temps. Cependant, puisque les feuilles de style offrent des moyens de présentation plus puissants, le W3C fera graduellement disparaître nombre d'éléments et d'attributs de présentation HTML. Les éléments et attributs concernés sont marqués comme « déconseillés ».
- **Scripts** : les scripts permettent aux auteurs de concevoir des pages Web dynamiques (par exemple, des « formulaires intelligents » qui réagissent au cours de leur remplissage par l'utilisateur) et d'employer HTML comme

support d'applications en réseau. Les mécanismes fournis pour associer HTML et scripts sont indépendants du langage de script.

- **Impression** : les auteurs voudront parfois aider l'utilisateur dans l'impression d'autres documents, en sus du document courant. Lorsque des documents font partie d'un ensemble, on peut décrire leurs relations en utilisant l'élément HTML `LINK` ou encore en utilisant le cadre de description des ressources (RDF) du W3C.

Fondamentalement, HTML 4 sépare bien plus efficacement la *structure* de la *présentation*. Les éléments et attributs de présentation HTML sont de plus en plus remplacés par d'autres mécanismes, en particulier les feuilles de style. Plus particulièrement, les anciens éléments `FONT` et `BASEFONT` sont désormais déconseillés.

La spécification HTML 4.01 est enfin une révision de HTML 4 qui corrige des erreurs et apporte certaines modifications à la version précédente. Vous pourrez trouver le texte en français de la spécification HTML 4.01 à l'adresse www.la-grange.net/w3c/html4.01/cover.html Cette spécification incorpore davantage d'emprunts au langage XHTML (*eXtensible HyperText Markup Language*), ce qui a permis d'alléger d'autant la spécification XHTML 1.0.

HTML 4.01 existe en trois « parfums ». Vous spécifiez la variante à employer en insérant une ligne au début du document. Chaque variante dispose de sa propre *définition de type de document*, ou DTD (*Document Type Definition*), qui spécifie les règles d'emploi de HTML de façon claire et succincte :

- Le DTD HTML 4.01 **strict** comprend tous les éléments et attributs qui ne sont pas déconseillés ou qui n'apparaissent pas dans les documents avec jeu d'encadrement. Pour les documents qui utilisent ce DTD, prendre la déclaration de type de document suivante :

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

- Le DTD HTML 4.01 **transitoire** inclut la totalité du DTD strict auquel se rajoutent les éléments et attributs déconseillés (la plupart d'entre eux concernant la présentation visuelle). Pour les documents qui utilisent ce DTD, prendre la déclaration de type de document suivante :

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

- Le DTD HTML 4.01 de **jeu d'encadrement** inclut la totalité du DTD transitoire complété des cadres (*frames*). Pour les documents qui utilisent ce DTD, la déclaration de type de document suivante est employée :

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

DTD

La définition de type de document, ou DTD (*Document Type Definition*), définit la structure d'un document, les éléments et attributs qui y sont autorisés, et le type de contenu ou d'attribut permis. Un document *bien formé* répond

simplement aux exigences de la spécification, tandis qu'un document *valide* se conforme strictement aux règles établies par la DTD à laquelle il fait référence.

L'étude complète des DTD dépasse l'objectif de ce livre. Une excellente présentation est disponible à l'adresse **http://developpeur.journaldunet.com/tutoriel/xml/031219xml_dtd1a.shtml**

HTML5 n'a plus recours aux DTD. Si l'instruction `DOCTYPE` reste nécessaire (mais pas en XHTML), elle se limitera à :

```
<!DOCTYPE html>
```

Dans l'ensemble de ce livre, nous nous conformerons à la spécification HTML 4.01.

1.2. Pendant ce temps…

Au cours de cette période, d'autres groupes de travail s'intéressaient à différents aspects plus ou moins intégrés progressivement dans les spécifications HTML successives.

XML

XML (*eXtensible Markup Language*) a été développé sous l'égide du W3C en 1996. Il décrit une classe d'objets nommés *documents XML* et décrit partiellement le comportement des programmes informatiques qui les traitent. XML est une application restreinte de SGML (*Standard Generalized Markup Language*, ISO 8879). Par construction, les documents XML sont des documents conformes SGML.

Ils sont constitués d'unités de stockage nommées « entités », renfermant des données analysées (*parsed*) ou non analysées (*unparsed*). Les données analysées sont composées de caractères, dont certains forment des données « caractère » et d'autres un balisage. Le balisage code une description de la structure logique de stockage du document. XML procure un mécanisme imposant certaines contraintes sur cette structure logique.

Un module logiciel nommé *processeur XML* sert à lire les documents XML et à procurer l'accès à leur structure et à leur contenu. Par définition, un processeur XML accomplit son travail pour le compte d'un autre module, l'application. Cette spécification décrit le comportement attendu d'un processeur XML : la façon dont il doit lire les données XML et les informations qu'il doit transmettre à l'application.

Les objectifs fondamentaux de XML sont les suivants :

- XML doit être largement utilisable sur Internet.
- XML doit prendre en charge une large gamme d'applications.

- XML doit être compatible avec SGML.
- L'écriture de programmes traitant les documents XML doit être facile.
- Les dispositifs facultatifs de XML doivent être limités voire, dans l'idéal, inexistants.
- Les documents XML doivent être clairs et lisibles par les humains.
- La conception XML doit pouvoir être effectuée rapidement.
- XML doit être formel et concis.
- Il doit être facile de créer des documents XML.
- Le côté abrupt du balisage XML importe peu.

Cette spécification, accompagnée de ses standards associés (Unicode et ISO/IEC 10646 pour les caractères, Internet RFC 3066 pour les balises d'identification de langue, ISO 639 pour les codes de noms de langue et ISO 3166 pour les codes de noms de pays), procure toutes les informations nécessaires pour comprendre XML Version 1.0 et construire les programmes informatiques qui le traitent.

CSS (Cascading Style Sheet)

Le langage CSS permet de définir des feuilles de style qui peuvent être appliquées à un site Web. Il permet la manipulation des styles appliqués à chaque balise HTML, *via* un langage de script.

La première Recommandation CSS (Recommandation W3C du 17 décembre 1996 révisée le 11 janvier 1999) a défini tout ce qui touche aux caractéristiques graphiques : couleurs, polices, tailles, etc.

Une seconde Recommandation (CSS2, 12 mai 1998, révisée le 8 avril 2008) a été ajoutée pour gérer tous les problèmes de positionnement dynamique. Elle définit les feuilles de style en cascade, niveau 2. CSS2 est un langage de feuille de style qui permet aux auteurs et aux lecteurs de lier du style (comme les polices de caractères, l'espacement et un signal auditif) aux documents structurés (comme les documents HTML et applications XML). En séparant la présentation du style du contenu des documents, CSS2 simplifie l'édition pour le Web et la maintenance d'un site.

CSS2 étant construit sur CSS1, toute feuille de style valide en CSS1 est ainsi également valide en CSS2, à quelques rares exceptions près. CSS2 prévoit des feuilles de style liées à un média spécifique, ce qui autorise les auteurs à présenter des documents sur-mesure pour les navigateurs visuels, les synthétiseurs de parole, les imprimantes, les lecteurs en braille, les appareils portatifs, etc. Cette spécification introduit aussi les notions de positionnement du contenu, de téléchargement des polices, de mise en forme des tables, de fonctions d'internationalisation, de compteurs et de numérotage automatique et quelques propriétés concernant l'interface utilisateur.

Aujourd'hui, un troisième projet (CSS3, Brouillon du 23 mai 2001) est en cours de réalisation. Il modularise CSS2. Plusieurs de ces modules sont

aujourd'hui à l'état de Candidats à la recommandation. Pour de plus amples informations, consultez le site du W3C.

À propos du DHTML

Le DHTML (*Dynamic HyperText Markup Language*) a été inventé par Netscape à partir de sa version 4. Ce n'est pas à proprement parler un langage de balises pour Internet : il n'existe d'ailleurs aucune norme DHTML à part entière. C'est en réalité un ensemble de technologies Internet associées afin de fournir des pages HTML plus dynamiques. Microsoft a suivi cette voie en développant une autre version de DHTML à partir de la version 4 d'Internet Explorer.

DHTML s'appuie sur HTML (nécessaire pour présenter le document), sur les feuilles de style (CSS), qui permettent de définir un style pour plusieurs objets et le positionnement de ceux-ci sur la page, sur le Modèle Objet de Document (DOM), proposant une hiérarchie d'objets afin de faciliter leur manipulation, et enfin sur un langage de script, essentiel pour définir des événements utilisateur. Il s'agit essentiellement de JavaScript (développé par Netscape) ou de JScript (développé par Microsoft), et éventuellement de VBScript, la tendance étant toutefois d'employer désormais ECMAScript, une tentative de normalisation du noyau du langage : sa syntaxe, ses mots-clés et ses composants natifs. La troisième édition du standard ECMA-262 a été publiée en décembre 1999 (www.ecma-international.org/publications/standards/Ecma-262.htm).

Sans script, le DHTML n'est pas dynamique. C'est l'association avec le script qui permet d'apporter des modifications après le chargement de la page, chose impossible avec le HTML classique. Avec ce dernier, une fois la page chargée et affichée, il est impossible d'afficher de nouveaux éléments ou de les déplacer.

Le DHTML serait très intéressant à utiliser s'il existait une norme officielle respectée par les navigateurs, ce qui reste loin d'être le cas : un script écrit pour un navigateur risque fort de ne pas fonctionner sur un autre sans travail d'adaptation. Même si chacun peut potentiellement profiter du DHTML, puisque pratiquement plus aucun internaute n'utilise de navigateur de génération inférieure à la version 4, les nombreuses incompatibilités entre navigateurs provoquent de grandes difficultés.

DHTML n'est donc en rien comparable au PHP, à l'ASP, aux CGI, qui permettent de formater « à la volée » les pages d'un site (souvent interfacé avec des bases de données) en proposant du contenu en temps réel et en interagissant avec l'utilisateur. La majorité des effets du DHTML restent ainsi réservés aux intranets, où la population des navigateurs est connue et maîtrisée.

DOM (Document Object Model)

Le *Modèle Objet de Document* (DOM level 2, Recommandation W3C du 13 novembre 2000) est la deuxième version d'une interface de programmation d'application (API) pour des documents HTML valides et XML bien formés. Il définit la structure logique d'un document et la manière d'y accéder et de le

manipuler. Dans la spécification DOM, le terme « document » est employé au sens large : XML sert de plus en plus à représenter de nombreuses sortes d'informations, qui peuvent être stockées sur des systèmes variés, et étaient traditionnellement considérées comme des données plutôt que comme des documents. XML présente néanmoins ces données comme des documents, le DOM permettant de gérer ces données.

Avec le Modèle Objet de Document, les programmeurs peuvent construire des documents, naviguer dans leur structure ainsi qu'ajouter, modifier ou effacer des éléments et leur contenu. Tout ce qui se trouve dans un document HTML, ou XML, peut être touché, modifié, effacé ou ajouté en utilisant le Modèle Objet de Document, à quelques rares exceptions près.

La recommandation DOM 3 Version 1.0 du 7 avril 2004 améliore DOM 2 en complétant la correspondance entre DOM et l'ensemble d'informations XML (*XML Information Set*), dont la prise en charge de XML Base. Cela rend possible l'ajout d'informations utilisateur aux nœuds DOM et fournit des mécanismes de résolution de préfixes d'espaces de noms ou de manipulation d'attributs id, etc. Le travail sur DOM se poursuit désormais au sein du groupe de travail sur HTML 5 et le prochain DOM devrait être nommé DOM 5.

Un objectif important du DOM en tant que spécification W3C est de fournir une interface de programmation standard pouvant être utilisée dans une grande diversité d'environnements et d'applications. Le DOM est conçu pour une utilisation dans n'importe quel langage de programmation.

1.3. La nouvelle génération : XHTML et HTML5

Après la mise au point de HTML 4, les membres du W3C ont délaissé HTML pour commencer à travailler sur un équivalent fondé sur XML, nommé XHTML (*eXtensible HyperText Markup Language*). La reformulation de HTML 4 en XML, nommée XHTML 1.0, a été achevée en 2000.

L'attention du W3C s'est alors portée sur l'extension de XHTML, sous l'égide de XHTML Modularization. Parallèlement, ils ont travaillé sur un nouveau langage incompatible avec les versions précédentes de HTML et de XHTML, nommé XHTML2.

2003 a vu l'apparition de Xforms. Cette technologie, présentée comme la nouvelle génération de formulaires Web, a suscité un nouvel intérêt pour l'évolution de HTML plutôt que son remplacement. Le déploiement de XML s'était en effet révélé limité à des technologies entièrement nouvelles (comme RSS et Atom), et non au remplacement de technologies existantes (comme HTML).

Il a alors été prouvé par Opera Software qu'il était possible d'étendre les formulaires HTML4 afin d'offrir la plupart des fonctionnalités offertes par XForms 1.0 sans imposer aux navigateurs de mettre en place de nouveaux moteurs de rendu incompatibles avec les pages Web HTML existantes.