

Cette spécification a été traduite par Michaël Muraz (banzaichico@yahoo.fr), dans un but pédagogique. Toute correction ou suggestion est la bienvenue.

Cette version française peut contenir des erreurs. Seule la version anglaise, disponible sur <http://www.w3.org/TR/xhtml1>, a une valeur normative.



XHTML™ 1.0 Le Langage de Balisage HyperTexte Extensible (Seconde Edition)

Une re-formulation de HTML 4 en XML 1.0

Recommandation du W3C du 26 janvier 2000, révisée le 1^{er} août 2002

Versions en Anglais :

Cette version :
<http://www.w3.org/TR/2002/REC-xhtml1-20020801>

Dernière version :
<http://www.w3.org/TR/xhtml1>

Version précédente :
<http://www.w3.org/TR/2000/REC-xhtml1-20000126>

Version avec corrections indiquées :
<http://www.w3.org/TR/2000/REC-xhtml1-20000126/xhtml1-diff.html>

Auteurs
Voir [Remerciements]

Se référer à l'[errata](#) de ce document qui doit inclure des corrections normatives. Voir aussi les [traductions](#).

Copyright ©2000 W3C® (MIT, INRIA, Keio), tous droits réservés. Les règles du W3C sur la responsabilité, les marques déposées, les droits d'auteurs et les licences de logiciels sont appliquées.

Résumé

Cette spécification définit la Seconde Edition de XHTML 1.0, une re-formulation de HTML 4 comme une application de XML 1.0 et trois DTD correspondant à celles définies par HTML 4. Les sémantiques de ces éléments et de leurs attributs sont définies dans la recommandation du W3C pour HTML 4. Ces sémantiques définissent les fondations de la future extensibilité de XHTML. Une comptabilité avec les agents utilisateurs HTML existants est possible en suivant un ensemble limité de règles.

Statut de ce document

Cette section décrit le statut de ce document à la date de sa publication. D'autres documents pourront remplacer ce document. Le dernier statut de la série de ce document est maintenu au W3C.

Ce document est la Seconde Edition de la spécification de XHTML 1.0 incorporant le errata des changements au 1^{er} août 2002. Les changements entre cette version et la précédente recommandation sont illustrés dans une version aux différences surlignées.

Cette seconde édition n'est *pas* une nouvelle version de XHTML 1.0 (publié pour la première fois le 26 janvier 2000). Les changements dans ce document reflètent les corrections appliquées comme un résultat des commentaires soumis par la communauté et comme un résultat de la suite des travaux à l'intérieur du Groupe de Travail HTML. Il n'y a pas de changements substantifs dans ce document - simplement l'intégration de différents errata.

La liste des erreurs connues dans cette spécification est disponible - en anglais - à l'adresse suivante : <http://www.w3.org/2002/08/REC-xhtml1-20020801-errata>.

S'il vous plaît, reportez les erreurs de ce document à www-html-editor@w3.org ([archives](#)). Une discussion publique sur les caractéristiques du HTML a lieu sur la mailing liste www-html@w3.org ([archives](#)).

Ce document a été produit dans le cadre de l'[Activité HTML du W3C](#). Les buts du Groupe de Travail HTML sont discutés dans la [charte du Groupe de Travail HTML](#).

A la date de publication, le groupe de travail pensait qu'il n'y avait aucune publication de brevet se rapportant à cette spécification. La liste courante des publications de brevets se rapportant à cette spécification peut être trouvée sur la [page des publication de brevets](#) du Groupe de Travail.

La liste des recommandations courantes du W3C et des autres documents techniques peut être trouvée à l'adresse suivante : <http://www.w3.org/TR>.

Table des matières rapide

1. Qu'est-ce que XHTML ?
2. Définitions
3. Définition Normative de XHTML 1.0
4. Différences avec HTML4
5. Problèmes de compatibilité
- A. DTD
- B. Interdictions d'Eléments
- C. Règles de Comptabilité HTML
- D. Remerciements
- E. Références

Table des matières complète

1. Qu'est-ce que XHTML ?
 - 1.1. Qu'est-ce que HTML 4 ?
 - 1.2. Qu'est-ce que XML ?
 - 1.3. Pourquoi XHTML ?
2. Définitions
 - 2.1. Terminologie
 - 2.2. Termes Généraux
3. Définition Normative de XHTML 1.0
 - 3.1. Conformité du Document
 - 3.1.1. Document Strictement Conformes
 - 3.1.2. Utiliser XHTML avec d'autres espaces de nominatif
 - 3.2. Conformités des agents utilisateurs
4. Différences avec HTML4
 - 4.1. Les documents doivent être bien rédigés
 - 4.2. Les éléments et les noms d'attributs doivent être en minuscules
 - 4.3. Les balise de fin sont requises pour les éléments non-vides
 - 4.4. Les valeurs des attributs doivent être entre guillemets
 - 4.5. La minimisation des attributs
 - 4.6. Les éléments vides
 - 4.7. La manipulation des espaces dans les valeurs des attributs
 - 4.8. Les scripts et les éléments de style
 - 4.9. Les exclusions SGML
 - 4.10. Les éléments avec les attributs « id » et « name »
 - 4.11. Les attributs avec des ensembles de valeurs prédéfinies
 - 4.12. Références aux entités par des valeurs hexadécimales
5. Problèmes de compatibilité
 - 5.1. Type de Média Internet
- A. DTD
 - A.1. Définitions de Type de Document
 - A.1.1. XHTML-1.0-Strict
 - A.1.2. XHTML-1.0-Transitionnal
 - A.1.3. XHTML-1.0-Frameset
 - A.2. Ensembles d'entités
 - A.2.1. Caractères Latin-1
 - A.2.2. Caractères spéciaux
 - A.2.3. Symboles

B. Interdictions d'Eléments

C. Règles de Comptabilité HTML

- C.1. Les Instructions de Développement et la Déclaration XML
- C.2. Elements Vides
- C.3. La Minimisation des Eléments et le Contenu des Eléments Vides
- C.4. L'Inclusion de Feuilles de Style et de Scripts
- C.5. Retours à la Ligne à l'intérieurs des Valeurs d'Attributs
- C.6. Isindex
- C.7. Les Attributs « lang » et « xml:lang »
- C.8. Les Identifiants Partiels
- C.9. L'Encodage des Caractères
- C.10. Les Attributs Booléens
- C.11. Le DOM et XHTML
- C.12. Utiliser « & » dans les Valeurs des Attributs (et Partout Ailleurs)
- C.13. Les CSS et XHTML
- C.14. Référencer les Eléments de Style lors d'une distribution sous XML
- C.15. Les Caractères Blancs en HTML opposé au XML
- C.16. La Référence de Caractère Nommée &apos ;

D. Remerciements

E. Références

1. Qu'est-ce que XHTML ?

Cette section est informative.

XHTML est une famille de types de document et de modules courants et futurs qui reproduit, détermine des sous-ensembles, et étend [HTML 4]. Les types de document de la famille XHTML sont basés sur XML, et est conçue finalement pour fonctionner en accord avec les agents utilisateurs basés sur XML. Les détails de cette famille et de son évolution sont discutés plus en détail dans [XHTMLMOD].

XHTML 1.0 (cette spécification) est le premier type de document dans la famille XHTML. C'est une re-formulation des trois types de document HTML 4 en tant qu'applications de [XML 1.0]. Il a été conçu dans le but d'être utilisé comme un langage pour le contenu qui est, à la fois, conforme à XML et, si des [règles] simples sont suivies, fonctionne également avec les agents utilisateurs compatible HTML 4. Les développeurs qui font migrer leur contenu vers XHTML 1.0 réaliseront les bénéfices suivants :

- Les documents XHTML sont conformes à XML. Ainsi, ils sont directement lisibles, éditables, et validables avec les outils XML standards.
- Les documents XHTML peuvent être écrits pour fonctionner aussi ou mieux qu'ils ne le faisaient précédemment dans les agents utilisateurs compatibles HTML 4 ainsi que dans les nouveaux agents utilisateurs compatibles XHTML 1.0.
- Les documents XHTML peuvent utiliser des applications (soit des scripts et des applets) qui repose sur le Modèle Objet de Document HTML autant que sur le Modèle Objet de Document XML [DOM].
- La famille XHTML des documents évoluant, les documents compatibles à XHTML 1.0 pourront interagir bien plus facilement au sein d'environnements XHTML variés.

La famille XHTML est la prochaine étape de l'évolution d'Internet. En migrant aujourd'hui vers XHTML, les développeurs de contenu peuvent entrer dans le monde XML avec tous ses bénéfices attendus, tout en restant confiant sur la compatibilité ascendante et future de leur contenu.

1.1 Qu'est-ce que HTML 4 ?

[HTML 4] est une application SGML (Standard Generalized Markup Language) conforme au standard international ISO 8879, et qui est largement admise comme le standard du langage de publication du World Wide Web.

SGML est un langage pour décrire les langages de structuration (balisage), particulièrement ceux utilisées dans l'échange de documents électroniques, la gestion documentaire, et la publication de document. HTML est un exemple de langage défini en SGML.

SGML a été créé au milieu des années 80 et est resté stable. Cette stabilité est inhérente au fait que le langage est riche de fonctionnalité et souple. Cette souplesse a, tout de même, un prix, et ce prix est un niveau de complexité qui a empêché son développement dans une grande diversité d'environnements, dont le World Wide Web.

HTML, comme il a été conçu à l'origine, a été défini pour être un langage d'échange de documents scientifiques et techniques, et utilisable par des non spécialistes de la grammaire syntaxique. HTML a résolu le problème de la complexité SGML en spécifiant un petit ensemble de balises sémantiques et structurelles, facilement utilisable pour l'écriture de documents relativement simples. De même pour simplifier la structure du document, HTML a ajouté la possibilité de l'hypertexte. Les capacités multimédia seront ajoutées plus tard.

En un espace de temps remarquablement court, HTML est devenu très populaire et a rapidement dépassé sa fonction première. Depuis le commencement de HTML, il y a eu rapidement création de nouveaux éléments au sein de HTML (en tant que standard) et pour adapter HTML aux marchés verticaux, hautement spécialisés. Cette pléthore de nouveaux éléments a finalement créé des disparités de compatibilité entre les différentes plateformes.

1.2 Qu'est-ce que XML ?

XML™ est l'abréviation de Extensible Markup Language [XML].

XML a été conçu comme un moyen de regagner la puissance et la souplesse de SGML sans la majorité de sa complexité. Bien qu'étant une forme restreinte de SGML, XML préserve pratiquement toute la puissance et la richesse de SGML, et respecte toutes les caractéristiques habituellement utilisées de SGML.

Bien que conservant ces options avantageuses, XML élimine la plupart des caractéristiques complexes de SGML qui rendent la conception et l'écriture de logiciels utiles à la fois difficile et coûteux.

1.3 Pourquoi XHTML ?

Les bénéfices de la migration vers XHTML 1.0 sont décrits plus hauts. Quelques uns des bénéfices en général sont :

- Les développeurs de document et les concepteurs d'agents utilisateurs ont constamment la possibilité de découvrir de nouvelles façons d'exprimer leurs idées à travers de nouveaux balisages. En XML, il est relativement facile d'introduire de nouveaux éléments ou des attributs d'éléments supplémentaires. La famille XHTML est conçue pour accepter ces extensions à travers les modules et les techniques XHTML pour développer de nouveaux modules conformes au XHTML (décrits dans la spécification sur la modularisation du XHTML). Ces modules permettront la combinaison d'ensembles de fonctionnalités existantes et futures tout en développant le contenu et en concevant les agents utilisateurs.
- Des manières alternatives d'accéder à Internet sont constamment créées. La famille XHTML est conçue en gardant à l'esprit la notion d'interopérabilité entre les agents utilisateurs. A travers un nouveau mécanisme de traitement de l'agent utilisateur et du document, les serveurs, les proxys et les agents utilisateurs seront capables d'effectuer la transformation la plus efficace possible du contenu. Finalement, il sera possible de développer du contenu compatible XHTML utilisable par n'importe quel agent utilisateur compatible XHTML.

2. Définitions

Cette section est normative.

2.1 Terminologie

Les termes suivants sont utilisés dans cette spécification. Ces termes étendent les définitions de [RFC2119] de façon basée sur les définitions similaires dans la norme ISO/IEC 9945-1:1990 [POSIX.1] :

Peut (may)

Dans le respect des implémentations, le mot « peut » (may) doit être interprété comme une caractéristique optionnelle qui n'est pas requise dans cette spécification mais peut être définie. Dans le respect de la [Conformité du Document], le mot « peut » (may) signifie que caractéristique optionnelle ne doit pas être utilisée. Le terme « optionnel » possède la même définition que « peut ».

Doit (must)

Dans la spécification, le mot « doit » doit être interprété comme une exigence obligatoire de l'implémentation ou des Documents Strictement Conformés au XHTML, ceci dépendant du contexte.

Optionnel (optional)

Voir « peut »

Réservé (reserved)

Une valeur ou un comportement n'est pas spécifié, mais son utilisation n'est pas autorisée dans les Documents Conformés et n'est pas non plus supportée par les Agents Utilisateurs Conformés.

Devrait (should)

Dans le respect des implémentations, le mot « devrait » (should) signifie une recommandation d'implémentation mais pas une obligation. Dans le respect des documents, le mot « devrait » (should) signifie une pratique de programmation recommandée pour les documents et requise pour les Documents Strictement Conformés.

Maintenu

Certaines facilités dans cette spécification sont optionnelles. Si un aménagement est maintenu, il se comporte comme spécifié par cette spécification.

Non-spécifié

Quand une valeur ou un comportement est non-spécifié, la spécification ne définit pas d'exigence de portabilité pour l'aménagement d'une implémentation même quand dans le cas d'un document qui utilise l'aménagement. Un document qui requiert un comportement spécifique pour une instance, plutôt que de tolérer des comportements lors de l'utilisation de cet aménagement, n'est pas un Document Strictement Conforme XHTML.

2.2 Termes Généraux

Attribut

Un attribut est un paramètre d'un élément déclarés dans la DTD. Un type d'attribut et intervalle de valeurs, incluant une possible valeur par défaut, sont définies dans la DTD.

DTD

Une DTD - ou Déclaration de Type de Document - est une collection de déclarations de balises XML qui, comme une collection, définit la structure, les éléments et les attributs légaux qui sont disponible pour l'usage dans un document qui est conforme à la DTD.

Document

Un document est un flot de données qui, après avoir été combinés avec tous les autres flots auxquels il fait référence, est structuré de telle façon qu'il comporte des informations contenues dans les éléments qui sont organisés selon leur définition dans la DTD associée. Voir la [Conformité du Document] pour plus d'informations.

Élément

Un élément est une unité de structuration de document déclarée dans la DTD. Le modèle de contenu de l'élément est défini dans la DTD, et les sémantiques supplémentaires doivent être définies dans le texte de description de l'élément.

Aménagements

Les aménagements sont des élément, des attributs et des sémantiques associées avec ces élément et attributs.

Implémentation

Voir Agent Utilisateur.

Parser

Parser est l'action par laquelle un document est scanné et l'information contenue dans le document est filtrée dans le contexte des éléments dans laquelle l'information est structuré.

Interprétation

L'interprétation est l'acte par lequel l'information d'un document est présentée. Cette présentation est faite dans la forme la plus appropriée à l'environnement (par exemple oralement, visuellement...)

Agent Utilisateur

Un agent utilisateur est un système qui traite les documents XHTML en accord avec cette spécification. Voir la [Conformité des Agents Utilisateurs] pour plus d'informations.

Validation

La validation est un procédé par lequel les documents sont vérifiés en fonction de la DTD associée, assurant que la structure, l'utilisation des éléments et l'utilisation des attributs sont conformes aux définitions dans la DTD.

Bien rédigé (Well-formed)

Un document est bien rédigé lorsqu'il est structuré en accord avec les règles définies dans la [section 2.1](#) de la Recommandation [XML 1.0].

3. Définition normative de XHTML 1.0

Cette section est normative.

3.1 Conformité de document

Cette version de XHTML fournit une définition des documents strictement conformes XHTML 1.0, qui sont restreints aux éléments et attributs des espaces de nominatifs du XML et du XHTML 1.0. Voir [section 3.1.2] pour information sur l'usage de XHTML avec d'autres espaces nominatifs.

3.1.1 Documents Strictement Conformes

Un document XHTML Strictement Conforme est un document XML qui requiert uniquement les aménagements décrits dans cette spécification. Un tel document doit répondre à tous les critères suivants :

- Il doit être conforme aux contraintes exprimées dans une des trois DTD que l'on peut trouver dans la section [DTD] et dans l'[Appendice B].
- L'élément racine du document doit être <html>.
- L'élément racine du document doit contenir une déclaration xmlns pour l'espace nominatif [XMLNS]. L'espace nominatif pour le XHTML est défini par <http://www.w3.org/1999/xhtml>. Un exemple de l'élément racine devra ressembler à ceci :

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
```

- Il doit y avoir une déclaration DOCTYPE dans le document avant l'élément racine. L'identifiant public inclus dans la déclaration DOCTYPE doit faire référence à une des trois DTD fournies dans la section [DTD] en utilisant l'Identifiant Public Formel correspondant. L'identifiant système peut être changé pour agréer aux conventions du système local.

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

- Le sous-ensemble de la DTD ne doit pas être utilisé pour outre-passer les paramètres de la DTD.

Une déclaration XML n'est pas requise dans tous les documents XML ; quoi qu'il en soit, les auteurs de documents XHTML sont fortement encouragés à utiliser dans tous leurs documents. Une telle déclaration est requise lorsque l'encodage des caractères du document est différente de celles par défaut UTF-8 ou UTF-16, et qu'aucun encodage n'est

déterminé par un protocole de niveau plus élevé. Voici un exemple d'un document XHTML. Dans cet exemple, la déclaration XML est incluse :

```
<?xml version="1.0" encoding="UTF-8"?>
<!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="en"
lang="en">
  <head>
    <title>Virtual Library</title>
  </head>
  <body>
    <p>Moved to <a
      href="http://example.org/">example.org</a>.</p>
  </body>
</html>
```

3.1.2 Utiliser XHTML avec d'autres espaces nominatifs

L'espace nominatif du XHTML peut être utilisé avec d'autres espaces nominatifs du XML comme pour [XMLNS] bien que de tels documents ne sont pas des documents strictement conformes XHTML1 comme défini ci-dessus. Le travail du W3C est de fournir des moyens de spécifier la conformité de documents comprenant plusieurs espaces nominatifs. Pour un exemple, voir [XHTML+MathML].

L'exemple suivant montre la manière par laquelle XHTML 1.0 peut être utilisé conjointement avec la Recommandation MathML :

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
  <head>
    <title>A Math Example</title>
  </head>
  <body>
    <p>The following is MathML markup:</p>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply> <log/>
        <logbase>
          <cn> 3 </cn>
        </logbase>
        <ci> x </ci>
      </apply>
    </math>
  </body>
</html>
```

L'exemple suivant montre la manière par laquelle une balise XHTML 1.0 peut être incorporée dans un autre espace nominatif XML :

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- initialement, l'espace nominatif par défaut est
"books" -->
<book xmlns='urn:loc.gov:books'
  xmlns:isbn='urn:ISBN:0-395-36341-6'
  xml:lang="en" lang="en">
  <title>Cheaper by the Dozen</title>
  <isbn:number>1568491379</isbn:number>
  <notes>
```

```
<!-- fait de HTML l'espace nominatif par défaut pour
un commentaire hypertexte -->
  <p xmlns='http://www.w3.org/1999/xhtml' >
    This is also available <a
      href="http://www.w3.org/">online</a>.
  </p>
</notes>
</book>
```

3.2 Conformité des agents utilisateurs

Un agent utilisateur conforme doit répondre aux critères suivants :

- Dans le but d'être compatible avec la recommandation [XML 1.0], l'agent utilisateur doit parser et évaluer un document XHTML pour sa bonne rédaction. Si l'agent utilisateur prétend être un agent utilisateur valideur, il doit aussi valider les documents par rapport à leur DTD référencées en accord avec [XML].
- Quand un agent utilisateur prétend supporter les aménagements définis dans cette spécification ou requis par cette spécification à travers une référence normative, il doit le faire par des moyens compatibles avec la définition des aménagements.
- Quand un agent utilisateur traite un document XHTML comme du XML générique, il doit seulement reconnaître les attributs de type ID (c'est-à-dire l'attribut id sur la plupart des éléments XHTML) en tant qu'identificateurs partiels.
- Si un agent utilisateur rencontre un élément qu'il ne reconnaît pas, il doit traiter le contenu de l'élément.
- Si un agent utilisateur rencontre un attribut qu'il ne reconnaît pas, il doit ignorer la spécification entière de l'attribut (c'est-à-dire l'attribut et sa valeur).
- Si un agent utilisateur rencontre une valeur d'attribut qu'il ne reconnaît pas, il doit utiliser la valeur par défaut de l'attribut.
- Si il rencontre une référence d'entité (autre qu'un de celles définies dans cette recommandation ou dans la recommandation XML) pour laquelle l'agent utilisateur n'a traité aucune déclaration (ce qui peut survenir si la déclaration est dans un sous-ensemble externe que l'agent utilisateur n'a pas lu), la référence devrait être traitée comme les caractères (commençant par « & » et finissant par le point-virgule) tel qu'ils décrivent la référence d'entité.
- Lorsque du traitement du contenu, les agents utilisateurs qui rencontrent des caractères ou des références d'entités de caractères qui son reconnus mais non interprétables peuvent substituer une autre interprétation qui donne la même signification, ou alors doivent afficher le document d'une telle manière qu'il est visible pour l'utilisateur que l'interprétation normale n'a pas eu lieu.
- L'espace blanc est manipulé en accords avec les règles suivantes. Les caractères suivants sont définis en [XML] comme des espaces blancs :
 - ESPACE ()
 - TABULATION HORIZONTALE ()
 - RETOUR CHARIOT()
 - FIN DE LIGNE(
)

Le processeur XML normalise différents systèmes de codes de fin de ligne en un seul caractère FIN DE LIGNE, qu'il passe ensuite à l'application.

L'agent utilisateur doit utiliser la définition venant de CSS pour traiter les caractères d'espacement [CSS2]. Notez que ce que la Recommandation CSS2 n'indique pas explicitement les sorties des manipulations d'espaces blancs dans les ensembles de caractères non-Latin. Cela sera indiqué dans une future version de CSS, à la date de laquelle cette référence sera mise à jour.

Notez que dans le but de produire un document XHTML Canonique, les règles ci-dessus doivent être appliquées et les règles dans [XMLC14N] doivent aussi être appliquées au document.

4. Différences avec HTML4

Cette section est informative.

Etant donné que XHTML est une application du XML, certaines habitudes qui étaient parfaitement légales en [HTML 4], basé sur SGML, doivent être changées.

4.1 Les Documents doivent être bien-rédigés

La rédaction correcte est un nouveau concept introduit par [XML]. Cela signifie essentiellement que tous les éléments doivent soit avoir une balise de fermeture, soit être écrits sous une forme spéciale (comme décrit ci-dessous), et que tous les éléments doivent être imbriqués correctement.

Bien que le chevauchement soit illégal en SGML, il est largement toléré dans les navigateurs existants.

CORRECT : Eléments imbriqués

```
<p>Voici un <em>paragraphe</em> en italique.</p>
```

INCORRECT : Eléments chevauchés

```
<p>Voici un <em>paragraphe en italique.</p></em>
```

4.2 Les noms des éléments et des attributs doivent être en minuscules

Les documents XHTML doivent être en minuscules pour tous les nom d'éléments et d'attributs. Cette différence est nécessaire car XML est sensible à la casse. Par exemple, `` et `` sont des balises différentes.

4.3 Les balises de fermetures sont requises pour les éléments non-vides

En HTML 4, basé sur SGML, il était permis d'omettre la balise de fin de certains éléments, en considérant que les éléments qui suivent créent une balise de fin implicite. XML ne permet pas l'omission des balises de fin. Tous les éléments autres que ceux déclarés dans la DTD comme `empty` (vide) doivent avoir une balise de fin. Les éléments déclarés dans la DTD comme `empty` (vide) peuvent avoir une balise de fin ou peuvent utiliser l'écriture rapide des éléments vides (voir [Eléments Vides]).

CORRECT : Eléments terminés

```
<p>Voici un paragraphe. </p><p>Voici un autre paragraphe. </p>
```

INCORRECT : Eléments non-terminés

```
<p>Voici un paragraphe. <p>Voici un autre paragraphe.
```

4.4 Les valeurs d'attributs doivent toujours être entre guillemets

Toutes les valeurs des attributs doivent être entre guillemets, même celles qui sont numériques.

CORRECT: Valeur d'attribut entre guillemets

```
<td rowspan="3">
```

INCORRECT: Valeur d'attribut sans guillemets

```
<td rowspan=3>
```

4.5 Minimisation des attributs

XML ne supporte pas la minimisation des attributs. La paire attribut-valeur doit être écrite en entier. Les noms d'attributs tels que `compact` et `checked` ne peuvent être placés dans des éléments sans que leur valeur soit spécifiée.

CORRECT: Attribut non-minimisé

```
<dl compact="compact">
```

INCORRECT: Attribut minimisé

```
<dl compact>
```

4.6 Éléments vides

Les éléments vides doivent soit avoir une balise de fin, soit la balise doit se terminer par « `/>` ». Par exemple `
` ou `<hr></hr>`. Voir les [Règles de Compatibilité HTML] pour information sur les moyens d'assurer une compatibilité avec les agents utilisateurs HTML 4.

CORRECT: Élément vides terminés

```
<br/><hr/>
```

INCORRECT: Élément vides non-terminés

```
<br><hr>
```

4.7 Manipulation des espaces dans les valeurs d'attribut

Quand les agents utilisateurs traitent les attributs, ils le font en accord avec la [Section 3.3.3](#) de [XML] :

- Recherche des espaces blancs.
- Regrouper les séquences de un ou plusieurs caractères d'espace blanc (incluant les retours à la ligne) en un seul espace entre les mots.

4.8 Les éléments de Style et de Script

En XHTML, les éléments de style et de script sont déclarés comme s'ils avaient un contenu #PCDATA. Par conséquent, « < » et « & » seront traités comme un début de balise et les entités telles que < et & seront reconnues respectivement comme des références aux entités « < » et « & » par le processeur XML. Encadrer le contenu de l'élément de script ou de style avec une section balisée CDATA évite la transformation de ces entités.

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

Les sections CDATA sont reconnues par le processeur XML et apparaissent comme des nœuds dans le Modèle Objet de Document, voir [Section 1.3](#) de la Recommandation DOM niveau 1 [[DOM](#)].

Une alternative est d'utiliser des documents de styles et des scripts externes.

4.9 Exclusions SGML

SGML donne le droit à l'auteur d'une DTD d'interdire certains éléments spécifiques d'être contenus dans un élément. De telles interdictions (appelées « exclusions ») ne sont pas possible en XML.

Par exemple, la DTD HTML 4 Stricte interdit l'imbrication d'un élément 'a' dans un autre élément 'a', quelque soit la profondeur. Ce n'est pas possible de définir de telles interdictions en XML. Même si ces interdictions ne peuvent être définies dans la DTD, certains éléments ne peuvent être imbriqués. Un résumé de tels éléments et des éléments qui ne doivent pas être imbriqués à l'intérieur peut être trouvé dans la section normative [Interdictions d'Eléments].

4.10 Les éléments avec les attributs id et name

HTML 4 définit l'attribut name pour les éléments a, applet, form, frame, iframe, img et map. HTML 4 introduit aussi l'attribut id. Ces deux attributs sont destinés à être utilisés comme des identifiants partiels.

EN XML, les identifiants partiels sont de type ID, et il peut y avoir seulement un attribut de type ID par élément. Par conséquent, en XHTML 1.0, l'attribut id est défini pour être de type ID. Dans le but d'assurer que les documents XHTML 1.0 sont des documents XML bien structurés, les documents XHTML 1.0 DOIVENT utiliser l'attribut id lors de la définition d'un identifiant partiel sur les éléments listés ci-dessus. Voir les

[Règles de compatibilité HTML] pour information sur l'assurance que de telles ancres sont compatibles lors de l'utilisation de documents XHTML en tant que média de type `text/html`.

Notez qu'en XHTML 1.0, l'attribut `name` des ces éléments est formellement déprécié, et sera supprimé dans une version ultérieure de XHTML.

4.11 Attributs avec ensembles de valeurs prédéfinies

HTML 4 et XHTML ont tous les deux des attributs qui ont des ensembles prédéfinis et limités de valeurs (par exemple l'attribut `type` de l'élément `input`). En SGML et XML, ils sont appelés *attributs énumérés*. Sous HTML 4, l'interprétation des ces valeurs était *insensible à la casse*. Sous XML, l'interprétation de ces valeurs est *sensible à la casse*, et en XHTML 1.0, toutes ces valeurs sont définies en minuscules.

4.12 Références d'entités en tant que valeurs hexadécimales

SGML et XML permettent tous deux les références aux caractères par utilisation de valeurs hexadécimales. En SGML, ces références peuvent être faites en utilisant soit `&#Xnn;`, soit `&#xnn;`. Dans les documents XML, vous devez utiliser la version en minuscules (`&#xnn;`).

5. Problèmes de Compatibilité

Cette section est normative.

Bien qu'il n'y ait pas d'obligation pour les documents XHTML 1.0 d'être compatibles avec les agents utilisateurs existants, en pratique, cela est facile à accomplir. Des règles pour créer des documents compatibles peuvent être trouvées dans l'[Appendice C].

5.1 Type de Média Internet

Les documents XHTML qui suivent l'ensemble des règles énoncées dans l'[Appendice C], « Règles de Compatibilité HTML », peuvent être nommés avec le Type de Média Internet « `text/html` » [RFC2854], puisqu'ils sont compatibles avec la majorité des navigateurs HTML. Ces documents, et n'importe quel autre document se conformant à cette spécification, peuvent aussi être nommés par le type « `application/xhtml+xml` » comme défini dans [RFC3236]. Pour plus d'information sur l'usage des types avec XHTML, voir la note informative [XHTMLMIME].

A. DTD

Cet appendice est normatif.

Ces ensembles d'entités et ces DTD forment une partie normative de cette spécification. L'ensemble complet des fichiers DTD avec une déclaration XML et un Catalogue SGML Ouvert est inclus dans le fichier zip et le fichier gzip'd tar de la version anglaise de cette spécification. Les utilisateurs recherchant des copies locales des DTD pour travailler avec devraient télécharger et utiliser ces archives plutôt qu'utiliser les DTD spécifiques référencées ci-dessous.

A.1 Définitions de Type de Document

Ces DTD approchent les DTD HTML 4. Le W3C recommande l'utilisation des versions à caractère autoritaire de ces DTD à leurs identifiants SYSTEME définies lors de la validation du contenu. Si vous avez besoin d'utiliser ces DTD localement, vous devriez télécharger une des archives de [cette version](#). Pour compléments, les versions normatives des DTD sont incluses ici :

A.1.1 XHTML-1.0-Strict

Le fichier DTD/xhtml1-strict.dtd est une partie normative de cette spécification. Le contenu annoté de ce fichier est disponible dans cette section séparée pour compléments.

A.1.2 XHTML-1.0-Transitionnal

Le fichier DTD/xhtml1-transitionnal.dtd est une partie normative de cette spécification. Le contenu annoté de ce fichier est disponible dans cette section séparée pour compléments.

A.1.3 XHTML-1.0-Frameset

Le fichier DTD/xhtml1-frameset.dtd est une partie normative de cette spécification. Le contenu annoté de ce fichier est disponible dans cette section séparée pour compléments.

A.2 Ensembles d'entités

Les ensembles d'entités XHTML sont les mêmes que pour HTML 4, mais ont été modifiés pour être valides avec les déclarations d'entités XML. Notez que l'entité pour le symbole Euro (€ ou € ou €) est définie comme faisant partie des caractères spéciaux.

A.2.1 Caractères Latin-1

Le fichier DTD/xhtml1-lat1.ent est une partie normative de cette spécification. Le contenu annoté de ce fichier est disponible dans cette section séparée pour compléments.

A.2.2 Caractères spéciaux

Le fichier DTD/xhtml-special.ent est une partie normative de cette spécification. Le contenu annoté de ce fichier est disponible dans cette section séparée pour compléments.

A.2.3 Symboles

Le fichier DTD/xhtml-symbol.ent est une partie normative de cette spécification. Le contenu annoté de ce fichier est disponible dans cette section séparée pour compléments.

B. Interdictions d'éléments

Cet appendice est normatif

Les éléments suivants ont des interdictions au niveau des éléments qu'ils peuvent contenir (voir [Exclusions SGML]). Cette interdiction s'applique à toutes les profondeurs d'imbrication, c'est-à-dire qu'elle concerne tous les éléments ascendants.

a

ne doit pas contenir d'autres éléments a.

pre

ne doit pas contenir les éléments `img`, `object`, `big`, `small`, `sub` ou `sup`.

button

ne doit pas contenir les éléments `input`, `select`, `textarea`, `label`, `button`, `form`, `fieldset`, `iframe` ou `isindex`.

label

ne doit pas contenir d'autres éléments `label`.

form

ne doit pas contenir d'autres éléments `form`.

C. Règles de compatibilité HTML

Cet appendice est informatif.

Cet appendice résume les règles de conception pour les auteurs qui souhaitent que leurs documents XHTML puissent être affichés sur les agents utilisateurs HTML existants. *Notez que cette recommandation ne définit pas comment les agents utilisateurs conformes au HTML doivent traiter les documents HTML. Elle ne définit pas non plus la signification du Type de Média Internet `text/html`. Pour ces définitions, voir respectivement [HTML 4] et [RFC2854].*

C.1 Les Instructions de Traitement et la Déclaration XML

Soyez conscient que les instructions de traitement sont affichées sur certains agents utilisateurs. De même, certains agents utilisateurs interprètent la déclaration XML comme signifiant que le document est du XML non reconnu plutôt que HTML, et par conséquent peuvent ne pas afficher le document comme attendu. Pour être compatible avec ces types de navigateurs, vous pouvez vouloir éviter d'utiliser les instructions de traitement et la déclaration XML. Souvenez-vous, de toutes façons, que lorsque la déclaration XML n'est pas incluse dans un document, celui-ci peut uniquement utiliser les encodages de caractères par défaut, UTF-8 et UTF-16.

C.2 Éléments vides

Incluez un espace avant les caractères « / » et « > » des éléments vides, par exemple `
`, `<hr />`, et ``. De même, utilisez la syntaxe des balises minimisées pour les éléments vides, par exemple `
`, puisque la syntaxe alternative `
</br>`, autorisée par XML, donne des résultats incertains sur de nombreux agents utilisateurs existants.

C.3 Minimisation des éléments et contenu des éléments vides

Soit une instance vide d'un élément dont le contenu n'est pas EMPTY (par exemple, un titre ou un paragraphe vide), n'utilisez pas la forme minimisée (par exemple, utilisez `<p> </p>` et non `<p />`).

C.4 Inclusion de Feuilles de Style et de Scripts

Utilisez des feuilles de styles externes, si votre feuille de style utilise « < » ou «]]> » ou « -- ». Utilisez des scripts externes, si votre script utilise « < » ou «]]> » ou « -- ». Notez que les parseurs XML sont autorisés à supprimer sans avertissement le contenu des commentaires. Par conséquent, la pratique historique de « cacher » des scripts et des feuilles de styles dans des commentaires pour faire en sorte que les documents soient compatibles avec les navigateurs anciens a des chances de ne pas fonctionner avec des agents utilisateurs basés sur XML.

C.5 Retours à la ligne dans les valeurs d'attribut

Évitez les retours à la ligne et les espaces blancs multiples à l'intérieur des valeurs d'attribut. Ceux-ci sont manipulés variablement par les agents utilisateurs.

C.6 `isindex`

N'incluez pas plus d'un élément `isindex` dans un document head. L'élément `isindex` est déprécié en faveur de l'élément `input`.

C.7 Les Attributs `lang` et `xml:lang`

Utilisez les deux attributs `lang` et `xml:lang` lors de la spécification de la langue d'un élément. La valeur de l'attribut `xml:lang` est prioritaire.

C.8 Identifiants partiels

En XML, les références URI [RFC2396] qui se terminent par un identifiant partiel de la forme `"#foo"` ne réfèrent pas aux éléments avec un attribut de la forme `name="foo"` ; ils réfèrent plutôt aux éléments avec un attribut défini pour être de type ID, par exemple l'attribut `id` de HTML 4. De nombreux clients existants ne supportent pas l'utilisation d'attributs de type ID de cette façon, des valeurs identiques peuvent donc être placées dans ces deux attributs pour assurer une compatibilité antérieure et ultérieure maximale (par exemple, `...`).

De plus, depuis que l'ensemble des valeurs légales pour les attributs de type ID est bien plus restreinte que celles de type CDATA, le type de l'attribut `name` a été changé en NMTOKEN. Cet attribut est tellement contraint qu'il peut uniquement avoir les mêmes valeurs que le type ID ou que la production `Name` dans la Section 2.3 de XML 1.0, production 5. Malheureusement, cette contrainte ne peut être exprimée dans les DTD XHTML 1.0. À cause de ce changement, il faut faire attention lors de la conversion de documents HTML existants. Les valeurs de ces attributs doivent être uniques dans le document, valides, et n'importe quelle référence à ces identifiants partiels (internes et externes) doivent être mises à jour.

Notez que la collection des valeurs légales dans la Section 2.3 de XML 1.0, production 5 est beaucoup plus étendue que ce qui est permis d'utiliser dans les types ID et NAME définis en HTML 4. Lors de la définition d'identifiants partiels pour être compatibles antérieurement, seules des chaînes de caractères vérifiant le pattern `[A-Za-z][A-Za-z0-9:_.-]*` doivent être utilisées. Voir [Section 6.2](#) de [HTML4] pour plus d'informations.

Finalement, notez que XHTML 1.0 a déprécié l'attribut `name` des éléments `a`, `applet`, `form`, `frame`, `iframe`, `img` et `map`, et sera supprimé de XHTML dans des versions ultérieures.

C.9 Encodage des Caractères

Historiquement, l'encodage des caractères d'un document HTML est soit spécifié par un serveur Internet via le paramètre `charset` de l'en-tête HTTP `Content-Type`, soit

via un élément `meta` dans le document lui-même. Dans les documents XML, l'encodage des caractères est spécifié par la déclaration XML (par exemple, `<?xml version="1.0" encoding="EUC-JP"?>`). Dans le but de la portabilité des documents présents avec des encodages spécifiques, la meilleure approche est de s'assurer que le serveur Internet définit l'en-tête correcte. Si ce n'est pas possible, un document qui veut placer ses propres encodages de caractères explicitement doit inclure à la fois une déclaration d'encodage dans la déclaration XML et un élément `meta http-equiv` (par exemple, `<meta http-equiv="Content-type" content="text/html; charset=EUC-JP" />`). Dans les agents utilisateurs conformes à XHTML, la valeur de la déclaration d'encodage XML est prioritaire.

Note : Soyez conscient que si un document doit inclure la déclaration d'encodage de caractères dans un élément `meta http-equiv`, ce document peut aussi être interprété par les serveurs http et/ou les agent utilisateurs comme étant de type média Internet définit dans cette déclaration. Si un document possède plusieurs types média, le serveur http doit avoir l'habitude de placer l'encodage de caractères du document.

C.10 Attributs Booléens

Quelques agent utilisateurs sont incapable d'interpréter les attributs booléens lorsqu'ils apparaissent dans le forme complète (non-minimisée), comme cela est requis par XML 1.0. Notez que ce problème n'affecte pas les agent utilisateurs conforme à HTML 4. Les attributs suivants sont concernés : `compact`, `nowrap`, `ismap`, `declare`, `noshade`, `checked`, `disabled`, `readonly`, `multiple`, `selected`, `noresize`, `defer`.

C.11 Le Modèle Objet de Document et XHTML

La recommandation du Modèle Objet de Document niveau 1 [DOM] définit des interfaces de modèles objet de document pour XML et HTML 4. Le modèle objet de document HTML 4 spécifie que les noms d'élément et d'attribut HTML sont retournés en majuscules. Le modèle objet de document XML spécifie que les noms d'élément et d'attribut sont retournés dans la casse avec laquelle ils sont spécifiés. En XHTML 1.0, les éléments et les attributs sont spécifiés en minuscules. La différence apparente peut être fixée de deux façons :

- Les agents utilisateurs qui accèdent à des documents XHTML distribués avec le type de média Internet `text/html` via le DOM peuvent utiliser le DOM HTML, et peuvent s'appuyer sur les noms d'élément et d'attribut retournés en majuscules par ces interfaces.
- Les agents utilisateurs qui accèdent à des documents XHTML distribués avec le type de média Internet `text/xml`, `application/xml`, ou `application/xhtml+xml` via le DOM peuvent aussi utiliser le DOM XML. De même, quelques éléments XHTML peuvent ou ne peuvent pas apparaître dans l'arbre objet car ils sont optionnels dans le contenu modèle (par exemple l'élément `tbody` dans `table`). Cela a lieu car en HTML 4 quelques éléments pouvaient tellement être minimisés que leurs balises de début et de fin étaient omises (une caractéristique SGML). Cela n'est pas possible en XML. Plutôt que de demander aux auteurs de documents d'insérer des éléments étrangers, XHTML a fait ces éléments optionnels. Les agent utilisateurs ont besoin de s'adapter à ceci. Pour plus d'informations sur ce sujet, voir [DOM 2].

C.12 Utiliser des & dans les valeurs d'attributs (et partout ailleurs)

Dans SGML et XML, le symbole « & » déclare le début d'une référence d'entité (par exemple, `®` pour le symbole ®). Malheureusement, de nombreux agents utilisateurs HTML ont ignoré sans avertissement l'usage incorrect de ce caractère dans les documents HTML, traitant les symboles « & » qui ne ressemblent pas à des références d'entités comme le symbole littéral. Les agents utilisateurs basés sur XML ne toléreront pas cet usage incorrect, et tout document qui utilise ce symbole incorrectement sera « invalide », et par conséquent non conforme à cette spécification. Dans le but d'assurer que les documents sont compatibles avec les agents utilisateurs historiques et les ceux basés sur le XML, les symboles « & » utilisés dans un document qui doivent être traités en tant que caractères littéraux doivent être exprimés en tant que référence d'entité (par exemple, `"&"`). Par exemple, lorsque l'attribut `href` d'un élément réfère à un script CGI qui prend des paramètres, il doit être exprimé comme ceci : `http://my.site.dom/cgi-bin/myscript.pl?class=guest&name=user` plutôt que comme cela : `http://my.site.dom/cgi-bin/myscript.pl?class=guest&name=user`.

C.13 Feuilles de Style (CSS) et XHTML

La recommandation CSS niveau 2 [CSS 2] définit des propriétés de style qui sont appliquées à l'arbre d'analyse grammaticale du HTML ou du XML. Des différences dans l'analyse produisent des résultats visuels ou oraux différents, dépendant de sélecteurs utilisés. Les indications suivantes réduiront cet effet pour les documents qui sont distribués sans modification des deux types de média :

- Les feuilles de style CSS pour XHTML devront utiliser des noms d'attribut et d'élément en minuscules.
- Dans les tableaux, l'élément `tbody` sera déduit par le parseur d'un agent utilisateur HTML mais pas par celui d'un agent utilisateur XML. Par conséquent, vous devrez toujours ajouter explicitement l'élément `tbody`, si un sélecteur CSS s'y réfère.
- Dans l'espace nominatif XHTML, les agents utilisateurs sont censés reconnaître l'attribut `"id"` comme un attribut de type `ID`. Par conséquent, les feuilles de style devront être capable de continuer à utiliser la syntaxe raccourcie de sélecteur `"#"` même si l'agent utilisateur ne lit pas les DTD.
- Dans l'espace nominatif XHTML, les agents utilisateurs sont censés reconnaître l'attribut `"class"`. Par conséquent, les feuilles de style devront être capable de continuer à utiliser la syntaxe raccourcie de sélecteur `"."`.
- CSS définit différentes règles de conformité pour les documents HTML et XML ; soyez conscient que les règles HTML s'appliquent aux documents XHTML distribués en tant que HTML et les règles XML s'appliquent aux documents XHTML distribués en tant que XML.

C.14 Référencer les Eléments de Style lors d'une distribution sous XML

En HTML 4 et XHTML, l'élément `style` peut être utilisé pour définir des règles de styles internes au document. En XML, une déclaration de feuille de style XML est utilisée

pour définir les règles de style. Dans le but d'être compatible avec cette convention, les éléments style devront avoir un identifiant partiel utilisant l'attribut id et une déclaration de feuille de style XML devra référencer cet identifiant partiel. Par exemple :

```
<?xml-stylesheet href="W3C-REC.css" type="text/css"?>
<?xml-stylesheet href="#internalStyle" type="text/css"?>
<!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="en"
lang="en">
<head>
<title>An internal stylesheet example</title>
<style type="text/css" id="internalStyle">
    code {
        color: green;
        font-family: monospace;
        font-weight: bold;
    }
</style>
</head>
<body>
    <p>
        Voici du texte qui utilise notre
        <code>feuille de style interne</code>.
    </p>
</body>
</html>
```

C.15 Les Caractères Blanc en HTML, opposé au XML

Quelques caractères qui sont légaux dans les documents HTML sont illégaux dans les documents XML. Par exemple, en HTML, le caractère « U+000C » est traité en tant qu'espace blanc, alors qu'en XHTML, à cause de la définition XML des caractères, ceci est illégal.

C.16 Les référence nommée de caractère '

La référence nommée de caractère ' (l'apostrophe, U+0027) fut introduite en XML 1.0, mais n'apparaît pas en HTML. Les auteurs devront par conséquent utiliser ' au lieu de ', comme attendu par les agents utilisateurs HTML 4.

D. Remerciements

Cet appendice est informatif.

Cette spécification a été écrite avec la participation des membres de Groupe de Travail HTML du W3C.

Lors de la publication de la seconde édition, la liste des membres était :

Steven Pemberton, CWI/W3C (HTML Working Group Chair)
Daniel Austin, Grainger
Jonny Axelsson, Opera Software
Tantek Çelik, Microsoft
Doug Dominiak, Openwave Systems
Herman Elenbaas, Philips Electronics
Beth Epperson, Netscape/AOL
Masayasu Ishikawa, W3C (HTML Activity Lead)
Shin'ichi Matsui, Panasonic
Shane McCarron, Applied Testing and Technology
Ann Navarro, WebGeek, Inc.
Subramanian Peruvemba, Oracle
Rob Relyea, Microsoft
Sebastian Schnitzenbaumer, SAP
Peter Stark, Sony Ericsson

Lors de la publication de la seconde édition, la liste des membres était :

Steven Pemberton, CWI (HTML Working Group Chair)
Murray Altheim, Sun Microsystems
Daniel Austin, AskJeeves (CNET: The Computer Network through July 1999)
Frank Boumphrey, HTML Writers Guild
John Burger, Mitre
Andrew W. Donoho, IBM
Sam Dooley, IBM
Klaus Hofrichter, GMD
Philipp Hoschka, W3C
Masayasu Ishikawa, W3C
Warner ten Kate, Philips Electronics
Peter King, Phone.com
Paula Klante, JetForm
Shin'ichi Matsui, Panasonic (W3C visiting engineer through September 1999)
Shane McCarron, Applied Testing and Technology (The Open Group through August 1999)
Ann Navarro, HTML Writers Guild
Zach Nies, Quark
Dave Raggett, W3C/HP (HTML Activity Lead)
Patrick Schmitz, Microsoft
Sebastian Schnitzenbaumer, Stack Overflow
Peter Stark, Phone.com
Chris Wilson, Microsoft
Ted Wugofski, Gateway 2000
Dan Zigmund, WebTV Networks

E. Références

Cet appendice est informatif.

[CSS2]

"*Cascading Style Sheets, level 2 (CSS2) Specification*", B. Bos, H. W. Lie, C. Lilley, I. Jacobs, 12 Mai 1998.

Dernière version - en anglais - disponible sur : <http://www.w3.org/TR/REC-CSS2>

[DOM]

"*Document Object Model (DOM) Level 1 Specification*", Lauren Wood et al., 1 Octobre 1998.

Dernière version - en anglais - disponible sur : <http://www.w3.org/TR/REC-DOM-Level-1>

[DOM2]

"*Document Object Model (DOM) Level 2 Core Specification*", A. Le Hors, et al., 13 Novembre 2000.

Dernière version - en anglais - disponible sur : <http://www.w3.org/TR/DOM-Level-2-Core>

[HTML]

"*HTML 4.01 Specification*", D. Raggett, A. Le Hors, I. Jacobs, 24 Décembre 1999.

Dernière version - en anglais - disponible sur : <http://www.w3.org/TR/html401>

[POSIX.1]

"*ISO/IEC 9945-1:1990 Information Technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C Language]*", Institute of Electrical and Electronics Engineers, Inc, 1990.

[RFC2045]

"*Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*", N. Freed and N. Borenstein, Novembre 1996.

Notez que cette RFC rend obsolètes RFC1521, RFC1522, et RFC1590.

[RFC2046]

"*RFC2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*", N. Freed et N. Borenstein, Novembre 1996.

Dernière version - en anglais - disponible sur : <http://www.ietf.org/rfc/rfc2046.txt>.

Notez que cette RFC rend obsolètes RFC1521, RFC1522, et RFC1590.

[RFC2119]

"*RFC2119: Key words for use in RFCs to Indicate Requirement Levels*", S. Bradner, Mars 1997.

Dernière version - en anglais - disponible sur : <http://www.ietf.org/rfc/rfc2119.txt>

[RFC2376]

"*RFC2376: XML Media Types*", E. Whitehead, M. Murata, Juillet 1998.

Ce document est rendu obsolète par [RFC3023].

Dernière version - en anglais - disponible sur : <http://www.ietf.org/rfc/rfc2376.txt>

[RFC2396]

"*RFC2396: Uniform Resource Identifiers (URI): Generic Syntax*", T. Berners-Lee, R. Fielding, L. Masinter, Août 1998.

Ce document met à jour RFC1738 et RFC1808.

Dernière version - en anglais - disponible sur : <http://www.ietf.org/rfc/rfc2396.txt>

[RFC2854]

"*RFC2854: The text/html Media Type*", D. Conolly, L. Masinter, Juin 2000.
Dernière version - en anglais - disponible sur : <http://www.ietf.org/rfc/rfc2854.txt>

[RFC3023]

"*RFC3023: XML Media Types*", M. Murata, S. St-Laurent, D. Kohn, Janvier 2001.
Ce document rend obsolète [RFC2376].
Dernière version - en anglais - disponible sur : <http://www.ietf.org/rfc/rfc3023.txt>.

[RFC3066]

"*Tags for the Identification of Languages*", H. Alvestrand, Janvier 2001.
Dernière version - en anglais - disponible sur : <http://www.ietf.org/rfc/rfc3066.txt>.

[RFC3236]

"*The 'application/xhtml+xml' Media Type*", M. Baker, P. Stark, Janvier 2002.
Dernière version - en anglais - disponible sur : <http://www.ietf.org/rfc/rfc3236.txt>

[XHTML+MathML]

"*XHTML plus Math 1.1 DTD*", "A.2 MathML as a DTD Module", Langage de balisage mathématique (MathML)
Dernière version - en anglais - disponible sur : <http://www.w3.org/TR/MathML2/dtd/xhtml-math11-f.dtd>

[XHTMLMIME]

"*XHTML Media Types*", Masayasu Ishikawa, 1 Août 2002.
Dernière version - en anglais - disponible sur : <http://www.w3.org/TR/xhtml-media-types>

[XHTMLMOD]

"*Modularization of XHTML*", M. Altheim et al., 10 Avril 2001.
Dernière version - en anglais - disponible sur : <http://www.w3.org/TR/xhtml-modularization>

[XML]

"*Extensible Markup Language (XML) 1.0 Specification (Second Edition)*", T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, 6 Octobre 2000.
Dernière version - en anglais - disponible sur : <http://www.w3.org/TR/REC-xml>

[XMLNS]

"*Namespaces in XML*", T. Bray, D. Hollander, A. Layman, 14 Janvier 1999.
L'espace nominatif XML définit une méthode simple pour qualifier les noms utilisés dans les documents XML en les associant avec des espaces nominatifs identifiés par URI.
Dernière version - en anglais - disponible sur : <http://www.w3.org/TR/REC-xml-names>

[XMLC14N]

"*Canonical XML Version 1.0*", J. Boyer, 15 Mars 2001.
Ce document décrit une méthode pour générer une représentation physique, la forme canonique, d'un document XML.
Dernière version - en anglais - disponible sur : <http://www.w3.org/TR/xml-c14n>