

Introduction à l'XHTML

par Adrien Pellegrini ([Page d'accueil](#))

Date de publication : 18/04/2007



Dernière mise à jour :

Cet article a pour but de vous expliquer dans les grandes lignes ce qu'est l'XHTML et d'expliquer les quelques différences et contraintes par rapport à l'HTML.

- 1 - Introduction à l'XHTML
 - 1.1 - Qu'est-ce que c'est ?
 - 1.2 - Quelle version utiliser, la 1.0 ou la 1.1 ?
- 2 - Validation W3C
 - 2.1 - Qu'est ce que c'est ?
 - 2.2 - Modalités de validation
 - 2.2.1 - DOCTYPE
 - 2.2.2 - Jeu de caractères
- 3 - Syntaxe
 - 3.1 - Bonne imbrication des balises
 - 3.2 - Nom et attribut des balises en minuscules
 - 3.3 - Balises "vides" avec terminaison
 - 3.4 - Balise de fermeture obligatoire pour les éléments non vides
 - 3.5 - Les attributs doivent être entre guillemets
 - 3.6 - Tout attribut doit avoir une valeur explicite
 - 3.7 - L'attribut "id" complémente "name"
- 4 - Ma première page XHTML
- 5 - Les balises
 - 5.1 - Définitions
 - 5.1.1 - Les éléments bloc et les éléments en-ligne
 - 5.1.2 - Les attributs
 - 5.1.3 - Un élément
 - 5.2 - Type : en-ligne
 - 5.2.1 - Balise <a> : liens
 - 5.2.2 - Balise
 : saut de ligne
 - 5.2.3 - Balise <cite> : citation
 - 5.2.4 - Balise <code>
 - 5.2.5 - Balise : mise en emphase
 - 5.2.6 - Balise : images
 - 5.2.7 - Balise <q> : citation
 - 5.2.8 - Balise
 - 5.2.9 - Balise : mise en emphase forte
 - 5.3 - Type : bloc
 - 5.3.1 - Balise <div> : division du document
 - 5.3.2 - Balises <h1> à <h6> : entête de paragraphe
 - 5.3.3 - Balise <hr /> : trait horizontal
 - 5.3.4 - Balise <p> : paragraphe
 - 5.3.5 - Balise <pre>
 - 5.3.6 - Balises et : liste
 - 5.4 - Un exemple complet
- 6 - Aller plus loin : passer de l'HTML à l'XHTML
 - 6.1 - Les balises obsolètes
 - 6.2 - Les obligations de l'XHTML
 - 6.2.1 - Attribut "alt" : texte alternatif
 - 6.2.2 - Encoder les caractères spéciaux
- 7 - Informations additionnelles
- 8 - Remerciements

1 - Introduction à l'XHTML

1.1 - Qu'est-ce que c'est ?

Le  **XHTML**, eXtensible HyperText Markup Language, est un langage identique au langage HTML mais respectant la syntaxe définie par  **XML**.

1.2 - Quelle version utiliser, la 1.0 ou la 1.1 ?

Pour l'instant, le **XHTML 1.0** convient car il est plus souple pour un passage de l'HTML à l'XHTML et certains navigateurs ne supportent pas le "content-type" du XHTML 1.1 .

Voici quelques différences entre la version 1.0 et la version 1.1 du XHTML :

- Un seul Document Type Definition DTD.
- Obligation d'utiliser le CSS pour le style du document.
- L'attribut *lang* est remplacé par l'attribut *xml:lang*.
- L'attribut *name* est remplacé par l'attribut *id* dans certaines balises.
- Prise en charge des éléments *ruby*.

2 - Validation W3C

2.1 - Qu'est ce que c'est ?

Le **W3C**, **World Wide Web Consortium**, est un organisme travaillant sur la mise en place de standards Web.

Ce qui nous intéresse plus particulièrement, ici, est leur système de validation de code source.

Par valider le code source, le système du W3C, va vérifier la syntaxe de votre code conformément aux règles établies.

Contrairement à ce que beaucoup pourraient penser, ce n'est pas dénué de sens de valider sa page ! En voici quelques raisons :

- Permet d'avoir un code portable, compatible avec la plupart des navigateurs.
- Permet d'avoir un code standardisé et connu de tout le monde.
- Permet une meilleure accessibilité.

2.2 - Modalités de validation

Pour avoir un site valide W3C, il faut :

- Renseigner le navigateur sur la norme utilisée via le DOCTYPE du document (ex: XHTML 1.0 STRICT).
- Renseigner le navigateur sur le jeu de caractères utilisé (ex: UTF-8).
- Se conformer à la syntaxe définie par le DOCTYPE utilisé (voir chapitre 3-Syntaxe pour plus d'informations)
- Passer un **test de validation W3C** du document XHTML

2.2.1 - DOCTYPE

Le **DOCTYPE**, Document Type, déclare au navigateur la norme utilisée.

Il doit être placé en tout début du code source, avant la balise `<html>`.

Pour le XHTML 1.0 :

XHTML 1.0 Strict :

XHTML 1.0 Strict

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

XHTML 1.0 Transitional :

XHTML 1.0 Transitional

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

XHTML 1.0 Frameset :

XHTML 1.0 Frameset

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



*Je vous conseille d'utiliser le **XHTML 1.0 Strict**.*

*Si par la suite vous souhaitez passer à l'**XHTML 1.1**, il sera bon de connaître l'**XHTML 1.0 Strict**.*

Différence entre les normes XHTML 1.0 :

- Strict : **utilisation recommandée !**
- Transitional : permet d'utiliser des balises "décoratives" comme , ... qui je le rappelle sont généralement dépréciées en XHTML Strict.
- Frameset : utilisé quand la page contient une ou plusieurs frames.

Pour le XHTML 1.1 :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

2.2.2 - Jeu de caractères

Pourquoi ?

Si le navigateur ne peut détecter le jeu de caractères de la page qu'il tente de visualiser, celui-ci affichera une page avec des caractères illisibles et incompréhensibles.

Spécifier le jeu de caractères est donc important !

Qu'est-ce que l'encodage de caractères?

L'encodage de caractères est un processus qui à chaque caractère associe un code que la machine peut comprendre.

Le codage le plus utilisé de nos jours est l'ISO-8859-1 qui permet d'encoder 191 caractères de l'alphabet latin.

Ensuite vient l'ISO-8859-15, qui permet, comme le précédent d'encoder l'alphabet latin mais permet également d'encoder des caractères fusionnés, €, ...

Aujourd'hui, l'internationalisation devient de plus en plus importante. Donc un jeu de caractères aussi restreint que l'ISO-8859-1 ne permet pas d'afficher des caractères dans d'autres langues tels que le japonais, etc. Pour pallier à ce problème, la norme Unicode a été inventée et donne place à l'UTF-8.

D'un point de vue général, peu importe l'encodage utilisé, dans un soucis d'interopérabilité, il vaut mieux toujours utiliser les entités HTML.

Il existe 2 méthodes qui peuvent être utilisées conjointement :

1er méthode :

```
<?xml version="1.0" encoding="utf-8" ?>
```

2ème méthode :

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

L'encodage UTF-8 est de plus en plus courant et a tendance à être recommandé.

Si toutefois vous ne voulez pas l'utiliser, vous pouvez choisir l'encodage actuel qui est le Latin 1 (ISO-8859-1).

Mais sachez que la table ISO-8859-1 ne contient que les caractères anglais, donc il vous faudra encoder les caractères absents de cet alphabet à l'aide d'entités HTML.

3 - Syntaxe

3.1 - Bonne imbrication des balises

Les balises doivent être correctement imbriquées selon leur hiérarchie.

Incorrect :

```
<h1>Vive Developpez.com !! Club d'entraide des développeurs <strong>francophones</h1><strong>
```

Correct :

```
<h1>Vive Developpez.com !! Club d'entraide des développeurs <strong>francophones</strong></h1>
```

3.2 - Nom et attribut des balises en minuscules

Les noms et les attributs des balises doivent s'écrire tout en minuscules.

Incorrect :

```
<H1>Vive Developpez.com !! Club d'entraide des <EM>développeurs</EM> francophones</H1>
```

Correct :

```
<h1>Vive Developpez.com !! Club d'entraide des <em>développeurs</em> francophones</h1>
```

3.3 - Balises "vides" avec terminaison

Les balises "vides" doivent se terminer par `/>`. Par balises "vides", j'entends les balises tels que `
`, `<hr />`, `<input ... />`,

Incorrect :

```
<input type="text">
```

Correct :

```
<input type="text" />
```

3.4 - Balise de fermeture obligatoire pour les éléments non vides

Les éléments non vides doivent se terminer obligatoirement par leur balise de fermeture. Par éléments "non vides", j'entends les éléments autres que `
`, `<hr />`,

Incorrect :

```
<p>Vive Developpez.com !! <p>Club d'entraide des développeurs francophones
```

Correct :

```
<p>Vive Developpez.com !!</p><p>Club d'entraide des développeurs francophones</p>
```

3.5 - Les attributs doivent être entre guillemets

Incorrect :

```
<p class=title>Vive Developpez.com !! Club d'entraide des développeurs francophones</p>
```

Correct :

```
<p class="title">Vive Developpez.com !! Club d'entraide des développeurs francophones</p>
```

3.6 - Tout attribut doit avoir une valeur explicite

Incorrect :

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

Correct :

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

3.7 - L'attribut "id" complémente "name"

L'attribut HTML `id` est utilisé par la fonction JavaScript `document.getElementById()`. C'est celui que nous utiliserons, plutôt que l'attribut `name` qui est destiné à tout langage côté serveur (mais qui peut avoir la même valeur, ce qui est d'ailleurs assez courant).

(extrait de : **Les formulaires et PHP5 de Guillaume Rossolini**).

Pour des raisons de compatibilité avec les anciens navigateurs, il est mieux d'utiliser encore les deux et non seulement l'attribut *id*.

L'attribut *name* sera supprimé dans les versions futures du XHTML concernant les balises a, applet, form, frame, iframe, img et map.

Correct :

```
<input type="image" alt="" src="dvp.png" name="dvp" />
```

4 - Ma première page XHTML

Voici un exemple très simple de page valide **XHTML 1.0 Strict**

Pour un encodage en UTF-8 :

```
1| <?xml version="1.0" encoding="utf-8"?>
2| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3|   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4| <html xml:lang="fr">
5| <head>
6|   <title>First Page</title>
7|   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8| </head>
9| <body>
10| <p>Hello World !</p>
11| </body>
12| </html>
```

Description du code :

L01 : Déclaration du type d'encodage du document.

L02-03 : Déclaration de la norme utilisée.

L04 : Définit l'environnement linguistique utilisé dans le document.

L06 : Titre de la page.

L07 : Autre déclaration du type d'encodage.

5 - Les balises

5.1 - Définitions

5.1.1 - Les éléments bloc et les éléments en-ligne

Les éléments bloc sont utilisés pour structurer un document et peuvent contenir d'autres éléments en bloc et/ou en-ligne, des données. Ils servent de *conteneur*.

Les éléments en-ligne sont utilisés pour donner une sémantique spécifique à quelques mots d'une phrase. Un élément en-ligne ne peut pas contenir un élément en bloc.

5.1.2 - Les attributs

Ce sont des propriétés qui caractérisent des balises.

Exemple l'attribut *name*, *id*, ...

Exemple où *name* et *type* sont des attributs

```
<input type="text" name="username" />
```

5.1.3 - Un élément

Un élément est l'ensemble formé par une balise, ses attributs et ses données.

En rouge : la balise

En vert : l'attribut

En noir : la donnée

Exemple d'élément :

```
<a href="http://www.developpez.com" > Developpez.com </a>
```

5.2 - Type : en-ligne

5.2.1 - Balise <a> : liens

Utilité :

Insère un lien ou un point d'ancrage. (1)

Exemple :

Un lien :

```
<a href="http://www.developpez.com">Developpez.com</a>
```

Un point d'ancrage :

```
<a href="#intro">Introduction</a>

<!-- Plus loin dans le code -->

<a id="intro"></a>
```

Aperçu :

[Lien vers l'aperçu.](#)

5.2.2 - Balise `
` : saut de ligne

Utilité :

Provoque un saut de ligne (balise souvent déconseillée).

Exemple :

```
Du texte
<br />
La suite ...
```

Aperçu :

[Lien vers l'aperçu.](#)

5.2.3 - Balise `<cite>` : citation

Utilité :

Insère une citation.

Exemple :

```
Quand le sage désigne la lune, l'idiot regarde le doigt.<cite>Proverbe chinois</cite>
```

Aperçu :

Lien vers l'aperçu.

5.2.4 - Balise <code>

Utilité :

Insère un bloc de code.

Exemple :

```
<code>
<?php
    echo 'Hello World !';
?>
</code>
```

Aperçu :

Lien vers l'aperçu.

5.2.5 - Balise : mise en emphase

Utilité :

Met en emphase du texte.

Exemple :

```
Vive Developpez.com et <em>moi</em> aussi !
```

Aperçu :

Lien vers l'aperçu.

5.2.6 - Balise : images

Utilité :

Insère une image.

Exemple :

```

```

Aperçu :

Lien vers l'aperçu.

5.2.7 - Balise <q> : citation

Utilité :

Insère une citation courte.

Exemple :

```
<q>Quand le sage désigne la lune, l'idiot regarde le doigt.</q>
```

Aperçu :

Lien vers l'aperçu.

5.2.8 - Balise

Utilité :

Couplée à du  **CSS**, la balise *span* met en forme du texte.

Exemple :

```
<span style="text-decoration:line-through">Du texte barré</span> et du texte normal.
```

Aperçu :

[Lien vers l'aperçu.](#)

5.2.9 - Balise `` : mise en emphase forte

Utilité :

Mise en emphase forte du texte.

Exemple :

```
Vive <strong>Developpez.com</strong>.
```

Aperçu :

[Lien vers l'aperçu.](#)

5.3 - Type : bloc

5.3.1 - Balise `<div>` : division du document

Utilité :

Joue le rôle de "conteneur" en permettant de structurer la page en plusieurs blocs.

Exemple :

```
<div>  
  <div>
```

```
    Le haut de page
</div>
<div>
    Le contenu principal de la page
</div>
<div>
    Le bas de page
</div>
</div>
```

Aperçu :

Lien vers l'aperçu.

5.3.2 - Balises <h1> à <h6> : entête de paragraphe

Utilité :

Définit un entête de paragraphe de niveau X (X pouvant aller de 1 à 6).

La balise <h1> ne doit être utilisée normalement qu'une seule fois.

Exemple :

```
<h1>Titre 1</h1>
<h2>Titre 2</h2>
<h3>Titre 3</h3>
<h4>Titre 4</h4>
<h5>Titre 5</h5>
<h6>Titre 6</h6>
```

Aperçu :

Lien vers l'aperçu.

5.3.3 - Balise <hr /> : trait horizontal

Utilité :

Insère un trait horizontal.

Exemple :


```
<p>Premier paragraphe</p>
<hr />
<p>Deuxième paragraphe</p>
```

Aperçu :

[Lien vers l'aperçu.](#)

5.3.4 - Balise <p> : paragraphe

Utilité :

Définit un paragraphe.

Exemple :

```
<p>Un paragraphe</p>
```

Aperçu :

[Lien vers l'aperçu.](#)

5.3.5 - Balise <pre>

Utilité :

Définit un bloc où le texte sera affiché tel qu'il figure dans le fichier source.

Exemple :

```
<pre>
  Les espaces sont pris en compte      !!!
  Yes !!
</pre>
```

Aperçu :

[Lien vers l'aperçu.](#)

5.3.6 - Balises et : liste

Utilité :

Définit une liste à puces.

La balise *ul* étant une liste à puces.

La balise *ol* étant une liste ordonnée.

Exemple :

```
<ul>
  <li>Menu</li>
  <li>Entrée</li>
  <li>Plat principal</li>
  <li>Plat secondaire</li>
  <li>Dessert (important)</li>
</ul>
```

Aperçu :

[Lien vers l'aperçu.](#)

5.4 - Un exemple complet

Voici un exemple complet d'une page XHTML qui est en rapport avec l'article :

Exemple de conception d'un site dynamique

```
<?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="fr" lang="fr-FR">
<head>
  <title>First Page</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
  <div id="menu_horizontal">
    <ul>
      <li><a href="http://localhost/site_dyna/index.php?id_page=2">Mon corps de rêve</a></li>
      <li><a href="http://localhost/site_dyna/index.php?id_page=3">Mes vacances</a></li>
      <li><a href="http://localhost/site_dyna/index.php?id_page=4">Mes loisirs</a></li>
      <li><a href="http://localhost/site_dyna/index.php?id_page=5">Me contacter</a></li>
```

```
</ul>
</div>

<div id="chemin_fer">
  <p>Vous êtes ici : -&gt; <a
href="http://localhost/site_dyna/index.php?id_page=1">Accueil</a></p>
</div>

<div id="bloc_central">
  <div id="menu_vertical">
    <ul>
      <li><a href="http://localhost/site_dyna/index.php?id_page=2">Mon corps de
rêve</a></li>
      <li><a href="http://localhost/site_dyna/index.php?id_page=3">Mes vacances</a></li>
      <li><a href="http://localhost/site_dyna/index.php?id_page=4">Mes loisirs</a></li>
      <li><a href="http://localhost/site_dyna/index.php?id_page=5">Me contacter</a></li>
    </ul>
  </div>

  <div id="contenu">
    <p>Accueil</p>
  </div>
</div>

<div id="pied_page">
  <hr />
  <p>Ceci est le pied de page...</p>
</div>
</body>
</html>
```

Aperçu :

Lien vers l'aperçu.

6 - Aller plus loin : passer de l'HTML à l'XHTML

6.1 - Les balises obsolètes

Il existe quelques balises dépréciées qui étaient utilisées dans l'HTML. Ces balises portaient sur l'aspect "présentation" d'une page web. En voici quelques unes :

Balise	Explication
applet	insère un applet Java
basefont	modifie la police utilisée (taille, couleur)
center	centre un élément
dir	utilisé pour les listes de répertoires
font	modifie le format du texte (police, taille, couleur)
isindex	insère un champ de texte
menu	insère une liste
s	barre du texte
stike	barre du texte
u	souligne du texte

6.2 - Les obligations de l'XHTML

6.2.1 - Attribut "alt" : texte alternatif

Pour des raisons d'accessibilité, il est désormais obligatoire de spécifier un texte alternatif qui décrit l'image en question. Ce texte alternatif s'affiche à la place de l'image quand celle-ci, pour une raison ou pour une autre, ne peut pas s'afficher. Une infobulle apparaît aussi lors du survol de l'image.

Incorrect :

```

```

Correct :

```

```

6.2.2 - Encoder les caractères spéciaux

Il est maintenant nécessaire d'encoder les caractères spéciaux en entités HTML.

Incorrect :

```
http://www.mon-site.com/index.php?page=home&id=3
```

Correct :

```
http://www.mon-site.com/index.php?page=home&id=3
```

Voici quelques autres caractères à encoder :

- 1 & (&)
- 2 < (<)
- 3 > (>)
- 4 (espace insécable)
- 5 " (")
- 6 ' (')

Les trois premiers étant les plus importants.

7 - Informations additionnelles

Livres :



XHTML et CSS



HTML et XHTML - La référence



Design web : utiliser les standards - CSS et XHTML

Liens :



FAQ : XHTML



Liste complète des balises XHTML



Div et CSS : une mise en page rapide et facile



Techniques de SEO : white hat



XHTML Media Types



Character sets and encodings in XHTML, HTML and CSS

8 - Remerciements

Tous mes remerciements à **Yogui**, **Kerod**, **Swöög** et **Bisûnûrs** pour leur relecture.

1 : Un point d'ancrage est un lien qui renvoie vers une autre partie de cette même page.