

Xhtml : déclaration et encodage des caractères

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
   <title>Déclaration et encodage des caractères</title>
  </head>
  <body>
  </body>
</html>
```

1 Déclaration

Cette déclaration permet d'indiquer au navigateur la manière d'interpréter le code html de la page. Le w3c recommande l'utilisation du xhtml pour réaliser des pages web, ce qui permet de rapprocher le html du langage de balises plus général : le xml. Le xhtml demande beaucoup plus de rigueur quant à l'écriture des balises. Par exemple, en html omettre la balise de fermeture de la balise <p> (</p>) ne provoque pas d'erreur alors qu'en xhtml toute balise d'ouverture doit obligatoirement être fermée. Certaines balises sont à la fois ouvrantes et fermantes comme br, dans ce cas la balises doit être écrite
 en xhtml et non
 (écriture html).

La première partie de la déclaration **html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"** indique donc que le document doit être interprété comme un fichier xhtml. Avec le terme **Strict** le navigateur ne doit avoir aucune «indulgence» si le code de la page ne respecte pas la norme xhtml.

La deuxième partie **"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"** correspond à l'adresse du fichier qui contient la norme xhtml pour que le navigateur puisse contrôler la conformité du code la page avec cette norme. Pour les curieux vous pouvez copier et coller cette adresse dans votre navigateur pour visualiser ce fichier. Cette dtd (Déclaration de Type de Document) indique par exemple que toute balise ouvrante doit avoir une balise fermante.

2 Espace de nom et langue

La balise ouvrante html comporte 2 couples attribut / valeur.

Le premier `xmlns="http://www.w3.org/1999/xhtml"` concerne l'espace de nom, notion propre au langage xml. En xml le choix des noms des balises étant libre, il peut arriver que le chargement de plusieurs fichiers xml provoque des conflits d'interprétations si certaines des balises qu'ils contiennent sont identiques. On utilise alors les espaces de noms pour fixer la «signification» des noms des balises. Dans notre cas la balise `<h1>` par exemple sera interprétée comme un titre principal alors qu'elle pourrait avoir un tout autre sens dans un autre fichier xml. Le deuxième couple attribut / valeur `xml:lang="fr"` indique simplement la langue utilisée dans le document.

3 Encodage des caractères

L'encodage des caractères permet au navigateur de gérer l'affichage des caractères spéciaux et accentués des différentes langues. Par défaut les caractères sont considérés par les navigateurs sous l'encodage ASCII qui correspond aux langues anglo saxonnes. Autant dire que les caractères accentués ne vont pas s'afficher correctement.

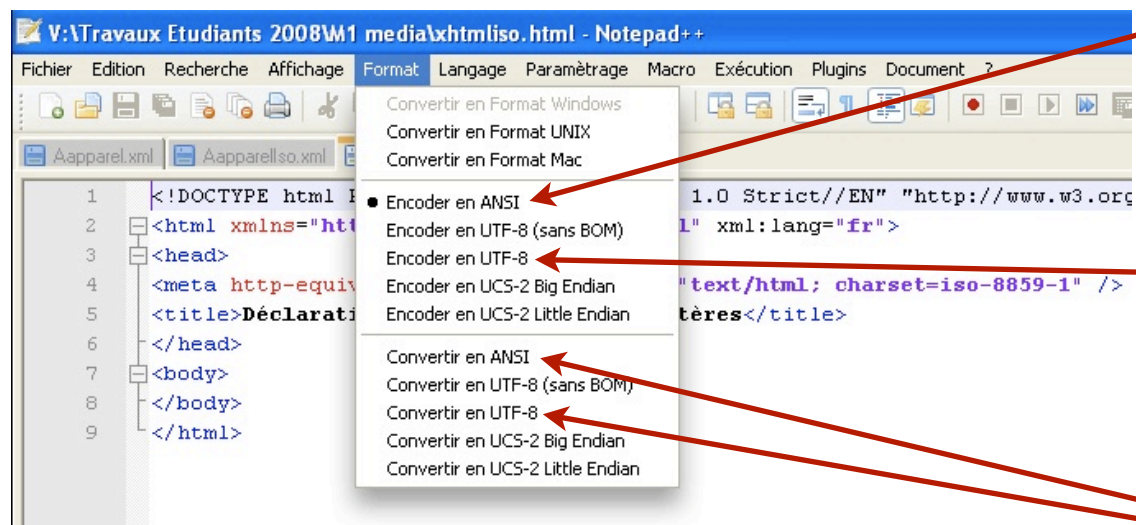
La première solution consiste à indiquer dans le code html l'entité du caractère accentué. Par exemple la lettre «é» s'écrit «é», le «è» s'écrit «è». Pour éviter la saisie de ces entités on utilise alors une balise `<meta>` dans l'en-tête pour spécifier au navigateur de faire lui-même cette conversion.

Cette balise s'écrit : `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />`.

La partie importante est la valeur de l'attribut «charset». En effet un ensemble de normes a été mise en place pour gérer l'ensemble des langues. Ainsi, pour les langues occidentales européennes la norme est iso-8859-1 (aussi appelée latin-1). On pourrait donc écrire : `<meta http-equiv="Content-Type" content="text/html; charset= iso-8859-1" />`.

La norme UTF-8 quant à elle permet de gérer de très nombreuses langues y comprises les langues asiatiques, arabes etc. Pour notre langue française on peut donc utiliser indifféremment iso-8859-1 ou UTF-8. D'autres critères (plus complexes à comprendre) peuvent intervenir pour choisir l'une de ces 2 normes, mais pour votre travail ces 2 normes fonctionnent.

ATTENTION : une fois que vous aurez choisi la norme (iso-8859-1 ou UTF-8) il faudra enregistrer vos fichiers html avec cet encodage. Dans notepad++, par exemple, vous trouverez ces encodages dans le menu «Format».



ANSI correspond à iso-8859-1. Sur cette capture d'écran le fichier est encodé en iso-8859-1, la balise meta sera donc : `<meta http-equiv="Content-Type" content="text/html; charset= iso-8859-1" />`

Si votre fichier est encodé en UTF-8 la balise meta devra être : `<meta http-equiv="Content-Type" content="text/html; charset= UTF-8" />`

Cette partie du menu format permet de transformer un encodage en un autre. Si vous faites une conversion n'oubliez pas d'enregistrer votre fichier.

Dans chacun des fichiers html que vous réaliserez devra apparaître les 3 éléments de codes que je vous ai détaillés : déclaration, espace de nom et langue et l'encodage des caractères.