Développer des composants d'interface

Créer des pages WEB
HTML - XHTML

Menu du jour

- Termes & acronymes
- Versions de HTML & XHTML
- Définition HTML & XHTML
- Différences en HTML & XHTML
- Environnement & balises
- Imbrication
- Paramètres
- Structure de base d'un document XHTML
- DTD
- Titres et paragraphes
- Séparateurs
- Balises et

Menu du jour

- L'attribut id
- Caractères accentués
- Les listes
- Les liens
- Les images
- Les images cliquables
- Les tables
- Validation d'un document XHTML
- Différences entre XHTML 1 & XHTML 2
- Liens sur internet

Termes & acronymes

CSS Cascading Style Sheets

DTD Document Type Definition

Document permettant de décrire un modèle de document XML

HTML Hypertext Markup Language

TD Table Data

W3C Word Wide Web Consortium

consortium fondé en octobre 1994 pour promouvoir la compatibilité des technologies du World Wide Web

XHTML Extensible HyperText Markup Language

Versions

- De HTML à XHTML
  - ⇨ HTML 1 (1992)
  - ⇨ HTML 2 (1994)
  - ⇨ HTML 3.2 (1997)
  - ⇨ HTML 4.x (1999)
  - ⇨ HTML 5.x (2008)
  - ⇨ XHTML 1.0 (26 jan 2000, revised 1 Aug 2002)
  - ⇨ XHTML 1.0 transitional
  - ⇨ XHTML 1.0 strict
  - ⇨ XHTML 1.0 frameset
  - ⇨ XHTML 1.1 - Module-based XHTML
  - ⇨ XHTML Basic 1.1 (version pour appareils portables)
  - ⇨ XHTML 2.x (sous préparation)

HTML : définition

- HTML est un langage de balises destiné essentiellement à la présentation des données
- HTML possède un ensemble de balises *fixé* destinées à :

  ⇨ structurer le document sous forme de titres, de paragraphes, de listes, de tableaux, ...

  ⇨ formater des données textuelles : emphase, citation, code, ...

  ⇨ définir des aspects de rendu visuel : police, taille et couleur des caractères, espacement et placement

XHTML : définition

- The Extensible HyperText Markup Language (XHTML) est un langage de balisage (“mark up”) qui définit la structure logique d’un document WWW diffusé sur le Web.
- XHTML est le pont entre les technologies HTML et XML
- Il vise à remplacer la première tout en se conformant à la seconde
- Pourquoi remplacer HTML ?

  ⇨ La syntaxe est HTML est très imprécise et très laxiste ce qui rend problématique
  - l'application de feuilles de styles CSS
  - l'utilisation d'applications (script/applet) pour rendre les pages et leur structure dynamique

  ⇨ Le langage de balises de HTML est fixé et n'a pas à vocation à être enrichi par un utilisateur

Différences en HTML & XHTML

- Différences majeurs entre HTML et XHTML
  - ⇨ Insertion d'un DOCTYPE correct au début du fichier (en principe aussi nécessaire pour HTML)
  - ⇨ Ajout d'un attribut de "namespace" au tag <html>
    xmlns="http://www.w3.org/1999/xhtml"
  - ⇨ Toutes les balises doivent être fermées
    par exemple ôter un n'est pas valable
  - ⇨ Lettres minuscules pour toutes les balises
    par exemple <P> devient
  - ⇨ Les éléments vides doivent avoir un "terminateur"
    par exemple <hr> devient <hr /> etc.
  - ⇨ Les valeurs d'attributs sont entourés de guillemets
    par exemple (vrai pour HTML)

Différences en HTML & XHTML

⇨ Chaque attribut doit avoir une valeur

par exemple <hr noshade="noshade">

⇨ Toujours utiliser & pour & dans des attributs, y compris les URL

par exemple <a href="?a=1&b=2">

⇨ La déclaration XML n'est pas nécessaire, mais fortement conseillée (surtout si vous utilisez des jeux de caractères comme ISO-8859-1 (accents)

<? xml version="1.0" encoding="ISO-8859-1" ?>

L'esprit XHTML

- Il convient de séparer le fond de la forme !
  - ⇨ Le fond :
    - Structuration des éléments d'un document : titre, paragraphe, liste, tableau, ...
    - Formatage de données textuelles de nature sémantique : emphase, citation, code, définition, ...
  - ⇨ La forme :
    - Aspects de rendu visuel : police, taille et couleur des caractères, ...
    - Espacement et placement sur la page des différents éléments

L'esprit XHTML

- Séparation fond/forme :

  ⇨ Le fond est le document XHTML

  ⇨ La forme est contenu dans une feuille de style CSS associée à la page XHTML
- Les outils de structuration du contenu d'un document doivent être utilisés en relation avec leur sémantique

  ⇨ Les listes ne sont pas des tableaux et inversement ...

  ⇨ Les tableaux **ne sont pas** un moyen de placement dans une page XHTML, ils ne servent qu'à structurer des éléments tabulaires

Environnement & balises

- Un **environnement** possède un début et une fin
- Un environnement est délimité par une **balise** ou marqueur (Angl. "marker" ou "tag") inséré au début et à la fin.
- Chaque marqueur est délimité par les signes < et >

  <balise> le contenu de l'environnement </balise>
- Les balises XHTML sont en lettres minuscules
- Exemple d'environnement XHTML

  ⇨ <h1>Titre principal</h1>

Imbrication

- Les environnements XHTML peuvent être imbriqués selon des règles bien définies
- Le chevauchement d'environnements n'est pas autorisé

```
JUSTE: <h1><a href="./test.html">Votre titre</a></h1>

FAUX !! <h1><a href="./test.html">Votre titre</h1></a>
```

Paramètres

- Les paramètres (ou attributs) modifient le comportement d'un environnement

  ⇨ Un paramètre se compose d'un mot clef (son nom) et d'une valeur

  ⇨ Les valeurs sont entre "guillements" (nom="valeur")
- Exemple qui définit le nom d'un fichier pour une image à inclure

  ⇨

Structure de base d'un document XHTML

```
// Déclaration XML
<?xml version = "1.0" encoding="UTF-8"?>

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

// Namespace declaration
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
<title>Object Model</title>
// Encoding declaration
<meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
</head>

// Contenu de la page
<body>
<h1>Cool cette page web</h1>
</body>
</html>
```

Structure de base d'un document XHTML

- Le document doit posséder une entête XML

  ⇨ Une déclaration XML (version et à option "jeu de caractères")

  ```
  <? xml version="1.0" encoding="ISO-8859-1"  ?>
  ```

  ⇨ Une déclaration du DocType ("XHTML")

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

- Le contenu du document doit être entouré de marqueurs (tags) "html" avec une déclaration de "namespace"

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

- L' en-tête est utilisé pour stocker de l'information à propos du document

Structure de base d'un document XHTML

- Le titre ("title") du document est utilisé à deux fins
  ⇨ Dans la plupart des "browsers", le titre est affiché dans la barre supérieure de la fenêtre
  ⇨ Le titre est souvent utilisé par les "robots" pour indexer votre page
- La page XHTML proprement dite (ce qui est visible dans le browser) est incluse dans le corps ("body")

DTD

- Le "Doctype", souvent appelé DTD (Déclaration de Type de Document) sert à indiquer à quelles règles d'écriture obéit le code d'une page HTML ou XHTML
- Ces règles sont en fait contenues dans des documents assez particuliers, les Définitions de Type de Document (abrégé également sous la forme DTD), écrites dans un langage un peu "barbare"
- Pourquoi appliquer de telles restrictions ?

  ⇨ La principale raison est de rendre l'application plus robuste. La connaissance d'informations définissant précisément les données qui seront lues permet d'éviter des erreurs et des incohérences
- Les DTD sont hébergées sur le site du W3C

- Vous disposez actuellement de 6 DTD
  - ⇨ HTML 4.01 transitional
  - ⇨ HTML 4.01 strict
  - ⇨ HTML 4.01 frameset
  - ⇨ XHTML 1.0 transitional
  - ⇨ XHTML 1.0 strict
  - ⇨ XHTML 1.0 frameset
- Qu'est-ce qui les différencie ?

  ⇨ Des règles de syntaxe HTML différentes : vous n'écrirez pas votre balisage de la même manière

  ⇨ Un stock de balises légèrement différent entre les trois catégories **transitionnelle**, **stricte** et **frameset**

  ⇨ La principale différence concerne les balises servant uniquement à créer des effets de présentation du texte. Ces effets peuvent être gérés de manière plus simple et plus souple à l'aide des styles CSS

- Trois types de DTD possibles :
  ⇨ transitional
  Un passage en douceur de HTML 4.01 vers XHTML 1.0
  ⇨ strict
  Restriction de XHTML 1.0 Transitional, beaucoup moins laxiste concernant l'esprit XHTML
  ⇨ frameset
  Pour faire des "frames" ....

HTML4.01 transitional

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

HTML4.01 strict

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

HTML4.01 frameset

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

XHTML1.0 transitional

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

XHTML1.0 strict

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

XHTML1.0 frameset

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

DTD

- XHTML1.0 ne sépare pas plus le contenu et la présentation qu'HTML 4.01 : dans les deux cas, c'est en fait le choix entre strict et transitional qui fait la différence
- Aucune de ces DTD n'apporte plus d'accessibilité : XHTML1.0 n'est pas plus accessible que HTML 4.01 C'est l'usage que vous en ferez qui fera la différence
- XHTML1.0 n'apporte aucun gain “sémantique” par rapport à HTML4.01, dont il reprend les éléments et la quasi-totalité des attributs. Là encore, c'est ce que vous en ferez qui fera la différence
- XHTML1.0 n'est pas plus difficile à apprendre qu'HTML4.01, au contraire la syntaxe rigoureuse limite les risques d'erreurs

- Liste des DTD recommandés par le W3C
  http://www.w3.org/QA/2002/04/valid-dtd-list.html

Caractères accentués

- Par défaut, il faut rentrer des codes spéciaux comme les accents, l'euro etc. en UTF-8 ou alors indiquer le jeu de caractères dans la déclaration XML

  Exemple : pour travailler avec les caractères européens ISO-8859-1

  <? xml version="1.0" encoding="ISO-8859-1" ?>

  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN« "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
- Par défaut tout fichier XML définit implicitement :

  <?xml version="1.0"?> est équivalent à

  <?xml version="1.0" encoding="UTF-8"?>
- Note : utilisez toujours *&* pour *&* (y compris dans des attributs, les URLs etc.):

  <a href="service.ch?a=1&b=2">

  au lieu de

  <a href="service.ch?a=1&b=2">

Titres et paragraphes

- Titres
  - ⇨ Il existe 6 niveaux de titres
  - ⇨ Un retour à la ligne est automatiquement inséré après un titre.
  - ⇨ Syntaxe: **<hn></hn>** où n est compris entre 1 et 6 (h1 est le plus grand titre possible)

```
<h1>Titre de niveau 1</h1>
```

- Paramètre
  - ⇨ **align** sert à spécifier l'alignement horizontal d'un titre
  - ⇨ Syntaxe: align="left ou center ou right ou justify"

```
<h1 align="center">Texte du titre centré</h1>
```

Paragraphes

- Paragraphe
  - ⇨ Le marqueur de paragraphe "**p**" produit un double retour de ligne.
  - ⇨ Syntaxe : ** ... **
- Paramètre
  - ⇨ **align** sert à spécifier l'alignement horizontal du paragraphe
  - ⇨ Syntaxe : align="left ou center ou right"

```
<p align="center">Cool ce paragraphe</p>
```

- Paragraphe en retrait
  - ⇨ Le marqueur "**blockquote**" décale le paragraphe à droite.
  - ⇨ Syntaxe : **<blockquote> ... </blockquote>**

```
<blockquote>Cool</blockquote>
```

Paragraphes

- Paragraphe préformaté

  ⇨ Le marqueur “**pre**" est utilisé lorsqu’on désire afficher du texte avec un caractère non proportionnel (“monospaced font”). Un “m” prend autant de place qu’un “i”.

  ⇨ Lorsqu’on insère plusieurs espaces à la suite, ceux-ci sont maintenus à l’affichage (ce qui n’est pas le cas avec le texte normal)

  Syntaxe : **<pre> ... </pre>**

Séparateurs

- Retour de ligne

  ⇨ Le marqueur “**br**” effectue un simple retour de ligne

  Syntaxe : **
**
- Paramètre

  ⇨ **clear** permet de forcer l’alignement du texte. Ce paramètre est particulièrement utile lorsqu’on l’utilise conjointement avec des images. Il introduit des retours de lignes jusqu’à ce que la marge droite, gauche ou les deux soient libres

  ⇨ Syntaxe: clear="left ou right ou all"

```
<br clear="left" />
<br clear="all" />
```

Séparateurs

- Ligne horizontale

  ⇨ Le marqueur “**hr**” insère une barre horizontale

  Syntaxe: <hr />

  ⇨ Paramètres de <hr>

  size : spécifie l’épaisseur de la barre horizontale en pixels. La valeur par défaut est 1

  Syntaxe: size="n"

  ```
  <hr size="3" />
  ```

  width : spécifie la longueur de la ligne, soit en pixels, soit en pourcentage de la largeur de la fenêtre

  Syntaxe: width="n"

  ```
  <hr width="200" />
  <hr width="60%" />
  ```

align : spécifie l'alignement de la barre

Syntaxe: align="left ou right ou center"

```
<hr align="center" />
```

noshade : lorsqu'il est présent dans le marqueur "hr" l'effet est une ligne pleine sans ombrage

Syntaxe: noshade="noshade"

```
<hr noshade="noshade" />
```

Balises et

- Les balises **** et ** ** ne servent qu'à une structuration du document pour le design
  ⇨ La balise est de type *inline*
  ⇨ La balise est de type *block*
- Elles s'utilisent avec les attributs id ou class
- Ces balises sont surtout utiles pour associer des styles

```
<span class="important">Attention</span>
<div id="menu_gauche">…..</div>
```

L'attribut id

- L'attribut id est commun à tous les éléments d'un document XHTML
- <body id="lecorps" ><li id="item4" >
- Règles :

  ⇨ Chaque valeur d'id ne doit apparaitre qu'une fois dans le document et un élément ne peut avoir qu'un seul id

  ⇨ La valeur d'id doit débuter par un souligné ou une lettre, le reste contenant souligné, lettre ou chiffre

  ⇨ Du fait de son unicité, la valeur de id est utilisée pour appliquer des règles de style à un élément précis, notamment pour les éléments et

Les listes

- Liste à puces
  ⇨ Syntaxe: <ul> ...

  Elément 1
  Elément 2

- Liste numérotée
  ⇨ Syntaxe: <ol> ...

  Elément 1 Elément 2

- Liste de définitions
  ⇨ Le marqueur **dl** délimite une zone de liste de définition qui contient des termes **dt** et des descriptions **dd**
  ⇨ Syntaxe: <dl> ... </dl>

Les listes

- Le marqueur **dt** introduit un terme (ou libellé) de définition utilisée au sein un élément <dl>

  ⇨ Syntaxe: <dt> ... </dt>
- Le marqueur **dd** introduit une description du terme de définition. Le résultat à l'écran est un décalage du texte vers la droite

  ⇨ Syntaxe: <dd> ... </dd>
- Exemple d'une liste de définitions

  <dl>

  <th>Titre de la liste</th>

  <dt>Terme 1</dt><dd>Définition du terme 1</dd>

  <dt>Terme 2</dt><dd>Définition du terme 2</dd>

  </dl>
- Combinaison de différents type de listes

  Les listes peuvent être emboîtées les unes dans les autres, il suffit pour cela de définir un élément d'une liste comme étant une autre liste

Les liens

- Les liens permettent de construire un hypertexte et peuvent être de différents types :

  ⇨ externes : un pointeur du document mène vers un autre document

  ⇨ internes : un pointeur renvoie à une section du même document

- Les marqueurs <a>

  ⇨ Un lien se définit par le marqueur <a ...> suivi du paramètre href="*URL*" qui définit l'adresse du document vers lequel le lien conduit. Il se termine par

  ⇨ Le texte ou l'image qui sont insérés entre les marqueurs de début et de fin sont les parties actives du lien et déclencheront le chargement du document lorsqu'on clique dessus

  Syntaxe: texte

  Site de l'AFPA

  Retour à l'accueil

Les liens internes et externes

- Les **liens externes** permettent de pointer vers un document référencé par une adresse URL ou par un chemin relatif.
- Ce document peut être un document HTML ou tout autre type de fichier (sons, images, etc ...)
- Il faut distinguer entre :

  ⇨ Une adresse absolue : on indique le chemin pour parvenir à la page cible en entier

  ⇨ Une adresse relative : on indique le chemin depuis la page courante
- Les **liens internes** permettent de construire des tables de matières et des renvois à l'intérieur d'un texte
- Un lien interne pointe vers une ancre, c'est à dire un endroit à l'intérieur d'un document défini par un nom

Les liens internes

- Il faut définir deux choses pour un lien interne
  ⇨ L'ancre interne
  Syntaxe :
  ⇨ Le lien vers l'ancre
  Syntaxe : texte

```
....
<a href="#para1">Paragraphe 1</a>
<a href="#para2">Paragraphe 2</a>
....
<a name=" para1"></a>
<h1>Partie 1</h1>
...
<a name=" para2"></a>
<h1>Partie 2</h1>
```

Les liens externes

- Liens externes vers d'autres protocoles/services Internet

  ⇨ Liens pour l'envoi d'un message e-mail et la lecture d'un newsgroup

  Un tel lien lance automatiquement l'application de messagerie électronique en incluant le nom du destinataire

  ⇨ Lien pour emmener le lecteur vers un groupe de discussion en mentionnant le nom de celui-ci dans la définition du lien

Lien pour l'envoi d'e-mail

```
<a href="mailto:bob.marley@jamaique.com">Envoyez moi un message</a>
```

Lien pour la lecture d'un newsgroup

```
Consultez le <a href="news:comp.info.systems.www.authoring.html">groupe de discussion sur l'édition en HTML</a>
```

Les liens externes

⇨ Liens pour le téléchargement d'un fichier depuis un serveur FTP

Ce type de lien est utilisé pour donner accès à un dialogue de téléchargement de fichiers

Le protocole indiqué dans l'URL est le protocole de transfert de fichiers ftp

Lien pour le téléchargement de fichiers

```
<a href="ftp://ftp.cesr.fr/">Acces au site FTP</a>
```

Les images

- Les principaux formats d'image affichables par les browsers sont *.gif, *.jpg et *.png
- Pour insérer des images à l'intérieur d'un document HTML, on utilise la commande "**img**"

  Syntaxe :
- Paramètres

  ⇨ Le paramètre "src" est obligatoire et contient un URL ou un chemin relatif vers un fichier de format .gif ou .jpg ou .png

  ⇨ Le paramètre "align" permet de spécifier l'alignement de l'image par rapport au texte (top, bottom, middle, right, left)

  ⇨ Le paramètre "alt" contient le commentaire que les personnes utilisant un browser textuel (sans images) voient à la place de l'image

Les images

⇨ Les paramètres “width” et “height” se réfèrent à la largeur et à la hauteur de l’image (en pixels).

⇨ Le paramètre “hspace” permet de spécifier la distance horizontale en pixels entre le texte environnant et le bord de l’image

⇨ Le paramètre “vspace“ permet de spécifier la distance verticale en pixels entre l’image et le texte qui l’entoure

⇨ Le paramètre “border” permet de créer un cadre autour de l’image. La largeur de la bordure est exprimée en pixels

Les images

- La question critique dans l'utilisation d'images est le temps de téléchargement nécessaire
- Les stratégies possibles pour accélérer le chargement

  ⇨ spécifier la taille de l'image dans la commande "img"

  ⇨ réduire la taille du fichier (attention : ce que l'on gagne en volume de fichier, on le perd en qualité d'image)
    - Choisir une **profondeur de couleur** inférieure de 8 bits lorsque vous sauvez l'image après traitement. La couleur de chaque point est encodée par un certain nombre de bits. 8 bits correspond à une image en 256 couleurs.
    - **Entrelacer** les images en format GIF. L'effet lors du chargement de l'image est que le lecteur voit apparaître une image de faible résolution qui s'améliore par plusieurs passages successifs.
- Pour les grandes images, utilisez une version de taille réduite de l'image dans votre document et mettez-y un pointeur vers la version normale de votre image

Les images cliquables

- Une image cliquable permet de présenter des menus graphiques en plus des liens textuels
- Définir une image cliquable revient à définir des zones sensibles (rectangulaires, circulaires ...) et des actions correspondant à chacune d'entre elles
- La définition d'une image cliquable comporte deux parties

  ⇨ La définition des zones et des actions à exécuter (map & area)

  ⇨ L'image et son lien vers la définition de zones (usemap)
- Le marqueur "map" sert à définir le début et la fin d'une section de définition de zones sensibles

  ⇨ Le paramètre "name" sert à définir un nom symbolique pour la définition de zones cliquables

  <map name=« namemap"> ... </map>

Les images cliquables

- Le marqueur **area** sert à définir des zones à l'intérieur d'une map

  ⇨ Le paramètre **shape** correspond à la forme de la zone sensible (shape = "rect ou poly ou circle")

  ⇨ Le paramètre **coords** indique les coordonnées en pixel pour définir la taille des zones

  Le coin en haut à gauche de l'image est le point d'origine

Pour les **rectangles** : les coordonnées à spécifier sont le coin en haut à gauche et celui en bas à droite

```
<area shape="rect" coords="16,17,249,77"  />
```

Pour les **cercles** : les coordonnées du centre et la valeur du rayon sont nécessaires (3 nombres)

```
<area shape="circle" coords="16,249,10"  />
```

Pour les **polygones** : les coordonnées pour chaque point doivent être spécifiées

```
<area shape="poly" coords="10,10, 10,15, 15,15, ...."/>
```

Les images cliquables

⇨ Le paramètre **href** contient l'URL qui est chargé lorsque l'utilisateur clique sur une des zones sensibles

Syntaxe: href="url" où URL est une adresse WWW

<area href="http://www.afpa.fr" />

```
<map name="mamap">
<area shape="rect" coords="16,17,249,77" href="http://www.afpa.fr" />
<area shape="rect" coords="16,91,249,213" href=" http://www.afpa.fr"  />
<area shape="rect" coords="16,228,249,437" href=" http://www.afpa.fr"  />
</map>
```

- Le paramètre **usemap** se met à l'intérieur d'une commande IMG et indique au client que l'image est cliquable
- La valeur du paramètre usemap contient l'adresse d'une section map qui peut se trouver dans le même document (auquel cas on le déclare en commençant par #) ou dans un autre document ("URL#nom_de_carte")

Exemple d'image cliquable

```
<map name=« logomap">
<area shape="rect" coords="16,17,249,77" href="http://www.afpa.fr" />
<area shape="rect" coords="16,91,249,213"
href=http://www.afpa.fr/afpa/qui_organisation.html  />
<area shape="rect" coords="16,228,249,437"  href=http://www.afpa.fr  />
</map>

<img src=« logo.gif" usemap="#logomap"  />
```

Les tables

- Les tables sont utiles pour :

  ⇨ Présenter des données numériques ou des tables de correspondance. Ceci correspond à l'utilisation traditionnelle des tableaux

  ⇨ Forcer la mise en page d'un document comportant plusieurs colonnes. Toutefois, il est préférable d'utiliser les CSS pour le positionnement
- Une table ("table") contient des lignes ("tr") qui contient à son tour des cellules/colonnes ("td")
- Le marqueur **table** définit le début et la fin d'une table

  ⇨ Syntaxe : <table> ... </table>
- Les paramètres du marqueur table :

  ⇨ Le paramètre **border** s'insère dans le marqueur de début de table et permet de spécifier la largeur des bordures

  Syntaxe : border="n" où n est un nombre de pixels

  <table border="2"> ... </table>

⇨ Le paramètre **cellspacing** permet de contrôler l'espacement entre deux cellules

Syntaxe: cellspacing="n" où n est un nombre de pixels

<table cellspacing="4"> ... </table>

⇨ Le paramètre **cellpadding** sert à fixer la distance se trouvant entre le bord d'une cellule et le texte qui s'y trouve

Syntaxe : cellpadding="m" où m est un nombre de pixels

<table cellpadding="4"> ... </table>

- Le paramètre **width** permet de forcer la largeur et la hauteur qu'occupe la table sur la page. La valeur n peut être exprimée en pixels ou en pourcentages

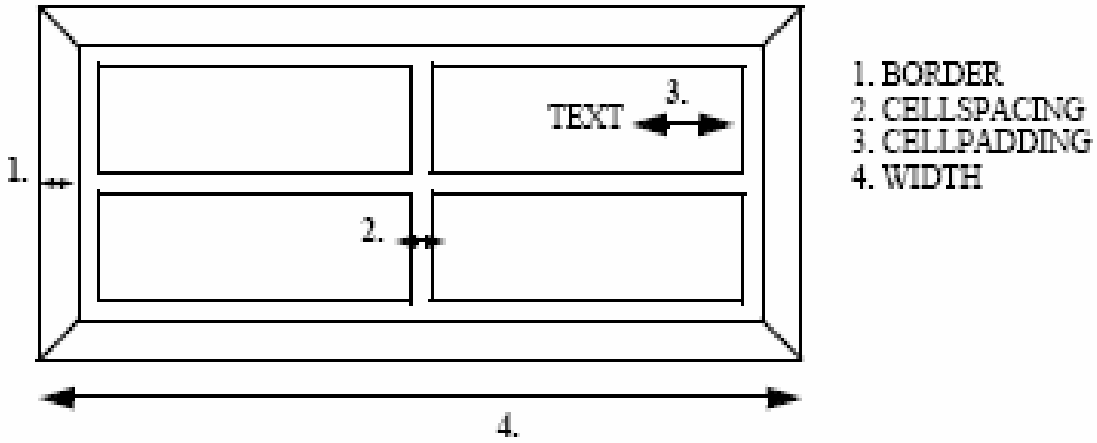
  Syntaxe: width="k"

  <table width="400"> ... </table>

  Syntaxe: width="l%"

  <table width="70%"> ... </table>

Résumé des paramètres du marqueur table



- Le marqueur **tr** définit le début et la fin d'une ligne du tableau

  Syntaxe: <tr> ... </tr>
- Paramètres du marqueur tr

  Ces paramètres s'appliquent aussi aux cellules td et th

⇨ Le paramètre **align** contrôle l'alignement horizontal du texte à l'intérieur de toutes les cellules d'une ligne s'il est spécifié dans un marqueur <tr> ou à l'intérieur d'une seule cellule s'il se trouve dans les marqueurs <td> ou <th>

Syntaxe: align="left ou center ou right"

<tr align="left" >

Toutes les colonnes contenues dans cette ligne seront alignées à gauche

⇨ Le paramètre **valign** contrôle l'alignement vertical du texte à l'intérieur des cellules

Syntaxe : valign="top ou middle ou bottom ou baseline"

<tr align="top" >

Toutes les colonnes contenues dans cette ligne seront alignées vers le haut

- Les marqueurs **td** et **th** servent à définir des cellules dans une ligne
- Le marqueur **td** définit le début et la fin d'une cellule
- Le marqueur **th** s'utilise comme **td** mais le résultat à l'affichage est un texte mis en évidence (apparaissant par exemple en gras)

  Syntaxe : <td> ... </td>
- Paramètres s'appliquant aux cellules :

  ⇨ Le paramètre **rowspan** détermine le nombre de lignes qu'une cellule occupe et s'introduit à l'intérieur des marqueurs <td> ou <th>

Syntaxe: rowspan="n" où n est un nombre de lignes

<td rowspan="2">

Une cellule qui occupe deux lignes

- Le paramètre **colspan** permet de définir une cellule qui occupe plusieurs colonnes. Le résultat est l'étirement de la cellule en largeur

  Syntaxe : colspan="m" où m est un nombre de cellules

  <td colspan="2">

Validation d'un document XHTML

- Pour vérifier que vos documents sont valides vis-à-vis de la DTD XTHML 1.0

  ⇨ Par l'intermédiaire du site :

  http://validator.w3.org/

  ⇨ En ajoutant à votre page le lien

  XHTML

Différences entre XHTML 1 & XHTML 2

- Principales différences :
  - ⇨ La balise **separator** remplace la balise hr
  - ⇨ La balise **l** remplace la balise br
  - ⇨ La balise **object** remplace la balise img
  - ⇨ Apparition des balises **section** et **nl**
- Détail des différences sur le site :

http://www.w3.org/TR/2006/WD-xhtml2-20060726/introduction.html#s_intro_differences

Liens sur internet

HTML 4.01 Specification
http://www.w3.org/TR/1999/REC-html401-19991224/

HTML 5 Working Draft 10 February 2008
http://www.w3.org/html/wg/html5/

W3C : XHTML™ Basic
http://www.w3.org/TR/2000/REC-xhtml-basic-20001219/

W3C : XHTML™ 2.0
http://www.w3.org/TR/2006/WD-xhtml2-20060726/