

# Partie III

## Du HTML au XHTML

### 1. Introduction

Depuis la création des premières pages Web (Par *TIM Berners-Lee* en 1990), les techniques de conception de ces dernières n'ont cessé d'évoluer. Aujourd'hui nous retrouvons une panoplie d'éditeurs manipulant du HTML et donc création des pages web de toutes formes. Ces pages Web sont interprétées par tous les navigateurs, c'est-à-dire que toute page récupérée sur Internet peut être visualisée sous un navigateur. Les navigateurs traduisent d'une manière transparente (pour l'utilisateur) le code HTML de la page en texte, images, sons,...

### 2. Création des pages Web avec HTML, XHTML

Depuis son apparition, le **HTML (Hypertext Markup Language)** est passé par plusieurs versions : HTML 1, HTML 2, HTML 3, HTML 4 puis HTML 5 (Finalisée par le « W3C » à la fin de l'année 2014). Ce langage a été le premier à utiliser pour la conception des pages Web. Dans ce chapitre nous nous intéresserons au langage **XHTML 1 (Extensible Hypertext Markup Language)** qui est apparu peu après le HTML 4. Ceci nous permettra d'étudier les deux langages (HTML et XHTML) au même temps, étant donné que dans XHTML il s'agit d'une reformulation plus rigoureuse du HTML. Seule la syntaxe change (elle est devenue plus précise, elle est également plus logique, et donc plus facile à comprendre), les fonctionnalités sont les mêmes.

### 3. Au cœur du Web ; Les éléments, les balises et les attributs

HTML et XHTML sont des langages de marquage (markup languages). Ils permettent d'enrichir un texte avec des informations structurelles, sémantiques et de présentation. Le principe d'un langage de marquage ou de balisage, consiste à utiliser des éléments délimités par des balises pour décrire des pages destinées au World Wide Web.

#### 3.1. Les éléments

Les éléments permettent d'associer aux différents blocs (texte, images...) les informations structurelles, sémantiques et de présentations désirées. Ils permettent de définir des titres, des paragraphes, des listes, des tableaux, des liens hypertextes, des images, etc...

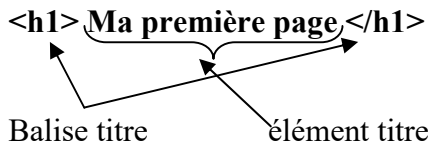
#### 3.2. Les balises

Les éléments sont délimités par des balises. On place une balise de début avant le contenu de l'élément et une balise de fin après. Mais dans certains cas, la balise de fin n'est pas nécessaire.

Une balise est formée en encadrant le nom de l'élément avec les symboles < et > (soit

<element> où element est le nom de l'élément). Pour une balise de fin on ajoute le caractère / avant le nom de l'élément (soit </element>).

### Exemple



### 3.3. Les attributs

Les attributs permettent d'apporter certaines précisions à des éléments. Il peut s'agir par exemple d'un nom de fichier (pour placer une image) ou d'une référence à une adresse HTML (lorsqu'on crée un lien). On place les attributs dans la balise de début de l'élément concerné. La syntaxe est simple : "nom de l'attribut"="valeur de l'attribut".

### 4. Règles pour l'écriture dans XHTML

Pour la conception d'une page web il faut suivre les règles suivantes :

- Chaque page web est un fichier dont le nom peut comprendre des lettres, des chiffres et des tirets (Il faut éviter les espaces, les caractères accentués), son extension est «.html» (pour des raisons techniques peut être « .htm »).
- Pour faciliter la navigation sur un site web, la première page du site doit être nommée index.html. En effet, c'est cette page de nom index qui s'affichera par défaut si l'internaute tape l'adresse du site web.
- Les balises s'écrivent toujours en minuscules (à l'inverse de HTML où on ne se soucie pas de la casse des balises).
- Chaque balise doit être refermée (sauf pour celles qui ne l'exigent pas).
- Ecrire les attributs des balises entre guillemets simples ou doubles.
- Il faut respecter l'imbrication de balises.

### 5. Structure d'une page XHTML

Pour bien comprendre la structure d'une page HTML ou XHTML voyons l'exemple « Ma première page » suivant:

```
<!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">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title> Ma première page </title>
  </head>
  <body>
    <h1>Bienvenue dans ma première page </h1>
    <h2>Faculté des SECSG </h2>
```

```

<p> Vous trouverez ici tout ce qui concerne mes enseignements, mes
projets, mes idées </p>
<ul>
  <li> Enseignements</li>
  <li> Projets </li>
  <li> Idées</li>
  <li> Autres</li>
</ul>
</body>
</html>

```

### Remarque

Pour dérouler cette exemple, il faut le recopier dans un nouveau document dans n'importe quel éditeur de texte (exemple bloc notes de Microsoft Windows) pour sauvegarder ce document avec l'extension «.html».

### Explication de l'exemple

**a. La balise** `<!DOCTYPE ...>` est le premier élément censé apparaître dans le code d'une page HTML. Elle permet au navigateur de savoir quelles règles appliquer pour la mise en page du document. Elle définit :

- La version HTML utilisée par le document.
- Le DTD (pour Document Type Definition ou Declaration) utilisé par le document.

Dans notre exemple `<!DOCTYPE ...>` nous indique que cette page est codée en XHTML Strict, version 1.0. Cette balise est trop compliquée à retenir, généralement on la colle à partir d'un modèle.

D'autres exemples peuvent être cités tels que :

#### HTML 4.01 Strict :

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

#### HTML 4.01 Transitional

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

#### HTML 4.01 Frameset

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

#### 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

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

#### XHTML 1.1

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

**Remarque**

Dans la dernière version du HTML « HTML 5 », la balise `<!DOCTYPE>` ne fait plus référence à un DTD et se déclare comme ceci :

```
<!DOCTYPE html>
```

La détection du type d'encodage du document HTML se fait en vérifiant d'autres balises dans le document HTML (par exemple la balise `<meta http-equiv="Content-Type">`).

**b. La balise `<html>` :** cette balise pourrait s'écrire simplement `<html>`, mais il est conseillé d'utiliser les deux attributs mentionnés ici : le premier (`xmlns : xml namespace`) donne un lien vers la liste des balises XHTML et le deuxième (`xml:lang`) précise la langue utilisée, ici « **fr** » pour le français. Là encore, un copier-coller s'impose pour cette deuxième et dernière balise un peu complexe.

**c. La balise En-tête `<head>` :** Elle contient les informations essentielles à la description du document qui sera affiché.

- La première ligne doit être nécessairement la balise `<meta ...>` qui indique le type de codage utilisé, ici **utf-8**.
- La balise `<title>` permet d'écrire un titre qui apparaîtra dans la barre de titre du navigateur.
- D'autres informations peuvent être ajoutées comme les références aux feuilles de style utilisées par le document, les références aux javascripts utilisés par le document.

**Remarque**

La balise `<meta ...>` peut servir à plusieurs utilisations dont:

\_ Fournir une description de la page : `<meta name="description" content=" ... " />`

Exemple : `<meta name="description" content="Technologies web"/>`

\_ Associer un ensemble de mots clés à la page : `<meta name="keywords" content=" ... " />`

`<meta name="keywords" content="XHTML, HTML, Web, XML"/>`

\_ Préciser le nom de l'auteur du document : `<meta name="author" content=" ... " />`

`<meta name="author" content="Mohamed Hamdi"/>`

**d. Corps de la page `<body>` :** Cette balise encadre le corps du document, c'est à dire les informations qui seront visibles dans le navigateur qui affichera la page : le texte, les images les liens, etc. Visuellement il correspond, donc, au cadre complet rendant l'affichage des pages du navigateur. Elle peut comporter les balises suivantes :

- La balise `<h1>` met en forme le titre de niveau 1 : grande taille, gras, espaces au-dessus et en dessous.

- La balise `<h2>` correspond à des sous-titres : taille un peu moins grande, un peu moins d'espacement autour.
- Le paragraphe `<p>` permet d'écrire une phrase.
- Enfin, la page se termine par une liste à puces, indiquée par la balise `<ul>`, chaque ligne étant repérée par une balise `<li>`, ce qui signifie retour à la ligne et nouvelle puce.

## 6. Principales balises du XHTML

### 6.1. Les commentaires

Il est très utile de placer des commentaires dans le code XHTML, pour s'y retrouver plus tard. Ils ne seront pas affichés par le navigateur, mais constituent une aide pour celui qui écrit et lit le code source de la page. Ils peuvent être écrits sur une ou plusieurs lignes et sont délimités par les balises `<!--` et `-->`.

#### Exemple

```
<!-- Ceci est un exemple -->
```

### 6.2. Paragraphes et titres

Chaque paragraphe est encadré par les balises `<p> ... </p>`. Un paragraphe ne contient jamais d'autres paragraphes, mais peut inclure des balises de mise en forme, comme le gras ou l'italique, des liens hypertexte et des images.

Les titres sont encadrés par les balises `<h1> ... </h1>` (**h** : header, en-tête en français). `<h1>`, `<h2>`, ... `<h6>` correspondent à différents niveaux de titre : `<h1>` est le premier niveau et avec le corps le plus grand, `<h2>` est le deuxième niveau et un texte légèrement plus petit, et `<h6>` est le sixième et dernier niveau dans la hiérarchie des titres et le plus petit texte.

### 6.3. Mise en forme courante de texte

La mise en évidence (par le changement de la mise en forme) de certains mots dans la page web s'obtient avec la balise `<em> ... </em>` qui signifie emphase (en anglais emphasis), c'est-à-dire accent ou insistance. Cette balise provoque, par exemple, la mise en italique des mots encadrés dans certains navigateurs et le soulignement dans d'autres. Dans tous les cas elle n'entraîne pas de retour à la ligne.

Pour mettre un terme en gras, il suffit de l'encadrer avec la balise `<strong> ... </strong>`, (stronger emphasis) ou avec la balise `<b> ... </b>` qui veut dire Bold.

Les textes en italique peuvent être définis par la balise `<i>... </i>` (pour italic), alors que le soulignement est défini par la balise `<u>... </u>` (pour underline).

Les textes barrés peuvent être définis par la balise `<s>... </s>`.

Les textes en indice peuvent être définis par la balise `<sub>...</sub>`, alors que les textes en puissance peuvent être définis par la balise `<sup>...</sup>`.

La balise `<br/>` (break) provoque un retour à la ligne ou retour chariot. Pour sauter plusieurs lignes, il est nécessaire de répéter la balise, celle-ci n'ayant pas d'attribut spécifiant le nombre de lignes à sauter. Il est possible d'utiliser la balise `<br/>` à l'intérieur d'autres balises (exemple `<p>`).

Des lignes horizontales peuvent être ajoutées aux paragraphes, elles sont définies par la balise unique `<hr/>`.

#### 6.4. Mise en forme commune à une partie du texte

Pour regrouper en un seul bloc un ensemble de paragraphes, de titres, ...etc., il suffit de les encadrer avec les balises `<div...> .... </div>` (div : une division ou partie d'une page).

##### Exemple

```
<div style="width:300px; height:120px; border:solid 1px black;">
... elements ....
</div>
```

##### Remarque

Dans l'exemple précédant l'attribut 'style' permet de définir un style de type CSS de la balise `<div>`.

Les balises `<span...> ... </span>` permettent de regrouper plusieurs mots d'un paragraphe et de leur donner une mise en forme commune, par exemple la taille, la couleur ou la police de caractère. Le mot anglais span signifiant portée, cela revient à sélectionner une partie du texte à mettre en forme, quelques mots ou quelques lettres.

##### Exemple

```
<p> Cette leçon est <span style="color:red;"> très très </span> importante </p>
```

#### 6.5. Liens hypertextes

La balise `<a href...>...</a>` permet d'écrire un lien hypertexte, le curseur prend la forme d'une main au passage de la souris sur ce lien, et un clic permet d'afficher la page qui est référencée dans cette balise (le lien).

L'attribut 'accesskey' permet d'activer le lien par un appui simultané sur la touche Alt et la lettre indiquée entre guillemets.

L'attribut 'title' permet d'afficher un texte automatiquement dans une bulle au passage de la souris sur le lien.

##### Exemple

```
<a href="http://www.google.fr" accesskey="a" title="Recherche sur google">
Chercher l'information sur google
</a>
```

Dans l'exemple précédant le lien fait référence à un objet externe (site google), il est donc appelé lien externe.

Un autre type de lien, appelé lien interne, fait référence vers un endroit (ou une ancre) de la page web courante. Pour mettre en place ce type de lien, il suffit d'ajouter un identifiant (ou une ancre) en attribut de la balise destination.

### Exemple

```
<a href="#en_bas"> Aller jusqu'en bas</a>
<a href="#au_centre"> Aller jusqu'au centre</a>
<br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
<a href="#" name="au_centre"> centre de la page </a>
<br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
<a href="#" name="en_bas"> bas de la page </a>
```

La combinaison de ces deux types permet de créer un lien vers un endroit précis d'une autre page que celle affichée.

### Exemple

```
<a href="Ma première page.html#3"> Mes idées </a>
```

Dans cet exemple le lien pointe sur la rubrique 3 (les idées) dans la page « Ma première page »

## 6.6. Listes à puces ou numérotées

Une énumération gagne en clarté lorsque chacun des points est repéré par une puce ou un numéro. De telles listes sont délimitées par les balises :

- `<ul>...</ul>` : pour les listes à puces (*unordered list*);
- `<ol>...</ol>` : pour les listes numérotées (*ordered list*);

Dans les deux cas, chaque ligne est repérée par `<li>...</li>` à l'intérieur de ces balises.

```
<p>Liste à puces :</p>
<ul>
  <li>Premièrement</li>
  <li>Deuxièmement</li>
  <li>Troisièmement</li>
  ...
</ul>
<p>Liste numérotée :</p>
<ol>
```

```

<li>Premièrement</li>
<li>Deuxièmement</li>
<li>Troisièmement</li>
...
</ol>

```

## 6.7. Tableaux

### a. Propriétés principales

En XHTML, un tableau est délimité par les balises `<table> ... </table>`. Initialement, les bordures du tableau sont invisibles, donc il faut ajouter dans la balise `<table>` l'attribut **border** pour indiquer l'épaisseur de la bordure en pixels.

La balise `<table>` peut inclure aussi les attributs **cellspacing** et **cellpadding** qui permettent de préciser en pixels, respectivement l'espacement entre cellules (traits d'encadrement simples si 0, doubles sinon) et l'espacement entre le texte et les bords des cellules.

Le tableau se construit ligne par ligne, avec les balises `<tr> ... </tr>` (*table row* : rangée du tableau). Les cellules du tableau sont définies à l'intérieur de chaque ligne, grâce aux balises `<td> ... </td>` (*table data* : donnée du tableau).

Lorsqu'il s'agit d'une cellule de titre, il est possible d'utiliser les balises `<th> ...</th>` (*table header* : en-tête du tableau) : le texte est alors mis en gras et centré.

### Exemple

Soit à écrire un tableau qui décrit les capitales des pays arabes :

```

<table border="1" cellspacing="0" cellpadding="10">
  <tr>
    <th>Pays</th>
    <th>Capitale</th>
  </tr>
  <tr>
    <td>Algérie</td>
    <td>Alger</td>
  </tr>
  <tr>
    <td>Tunisie</td>
    <td>Tunis</td>
  </tr>
</table>

```



## b. Autres propriétés des tableaux

La balise `<caption> ... </caption>` peut être ajoutée sous la balise de début de tableau `<table>`, elle permet de définir un titre associé au tableau, qui s'écrira par défaut au-dessus de celui-ci.

On peut spécifier la largeur du tableau avec l'attribut **width** placé dans la balise **table**. Cette valeur sera indiquée en pixel (exemple : **width="200"**) ou en pourcentage (exemple : **width="100%"**).

Aussi, pour chaque cellule, on peut spécifier une largeur à l'aide de l'attribut **width** placé dans la balise `<td>` ou `<th>`. De même que pour la largeur du tableau, la largeur de cellule peut être spécifiée en pixel (exemple : **width="200"**) ou en pourcentage (exemple : **width="100%"**). Cependant il faut faire attention :

- à la cohérence des largeurs données à chaque cellule. Si on utilise des pourcentages, le total doit être égal à 100%, si on utilise des pixels, le total ne doit pas dépasser la largeur du tableau.
- Il n'est pas utile de préciser la largeur de chaque cellule d'une colonne. On donne généralement une largeur à la première cellule d'une colonne et celle-ci sera valable pour toute la colonne.

Pour chaque cellule, on peut aussi préciser l'alignement horizontal du texte à l'aide de l'attribut **align** placé dans la balise `<td>` ou `<th>` et qui peut prendre les valeurs suivantes :

- left (texte aligné à gauche)
- center (texte aligné au centre)
- right (texte aligné à droite)
- justify (texte justifié, peut utiliser sur le web car il n'a pas de bon rendu)

Pour chaque cellule, on peut préciser l'alignement vertical du texte à l'aide de l'attribut **valign** (vertical align) placé dans la balise `<td>` ou `<th>` et qui peut prendre les valeurs suivantes :

- top (texte aligné en haut)
- middle (texte aligné au milieu)
- bottom (texte aligné en bas)

### Exemple récapitulatif

Reprenant le tableau qui décrit les capitales des pays arabes :

```
<table border="1" cellspacing="0" cellpadding="10" width="500">
```

```
<caption> Liste des pays arabes </caption>
```

```
<tr>
```

```
<th width="200" align = "center" valign = "middle" >Pays</th>
```

```
<th width="200" align = "center" valign = "middle" >Capitale</th>
```

```

<th width="100" align = "center" valign = "middle" >Population en
millions </th>
</tr>
<tr>
<td align = "center" valign = "middle">Algérie</td>
<td align = "center" valign = "middle">Alger</td>
<td align = "right" valign = "middle">35</td>
</tr>
<tr>
<td align = "center" valign = "middle">Tunisie</td>
<td align = "center" valign = "middle">Tunis</td>
<td align = "right" valign = "middle">6</td>
</tr>
</table>

```

### c. Fusionner des cellules

L'attribut **colspan** permet de fusionner horizontalement les cellules de plusieurs colonnes. Par exemple, la balise `<td colspan="3">...</td>` sera équivalente à trois cellules `<td>...</td>` à l'intérieur d'une ligne.

De même, l'attribut **rowspan** sert à fusionner verticalement les cellules de plusieurs lignes. Une balise `<td rowspan="4">...</td>` sera équivalente à quatre cellules dans le sens vertical ce qui signifie que les trois lignes qui suivent auront une cellule `<td>...</td>` en moins.

### Exemple

Reprenant le tableau qui décrit les capitales des pays arabes :

```

<table border="1" cellspacing="0" cellpadding="10" width="500">
<caption> Liste des pays arabes </caption>
<tr>
<th rowspan = "2" >Pays</th>
<th colspan = "3"> Caractéristiques</th>
</tr>
<tr>
<th >Capitale</th>
<th >Population en millions </th>
<th >Superficie en km2 </th>
</tr>

```

```

<tr>
  <td >Algérie</td>
  <td >Alger</td>
  <td >35</td>
  <td >2300000</td>
</tr>
<tr>
  <td >Tunisie</td>
  <td >Tunis</td>
  <td >6</td>
  <td >230000</td>
</tr>
</table>

```

## 6.8. Codage des caractères spéciaux

Le tableau suivant donne les séquences relatives au codage des caractères spéciaux :

&nbsp;	Espace insécable	&brvbar;		&plusmn;	±
&pound;	£	&yen;	¥	&Oslash;	Ø
&copy;	©	&reg;	®	&deg;	°
&sup2;	<sup>2</sup>	&sup3;	<sup>3</sup>	&amp;	&
&frac14;	¼	&frac12;	½	&frac34;	¾
&agrave;	à	&Agrave;	À	&oelig;	œ
&eacute;	é	&Eacute;	É	&euml;	ë
&icirc;	î	&Icirc;	Î	&ccedil;	ç
&quot;	"	&szlig;	ß	&ntilde;	ñ
&lt;	<	&gt;	>	&micro;	µ

### 7.1. La balise Liste de définitions <dl>...</dl>

La balise liste de définitions est utilisée pour organiser les définitions de plusieurs termes (une série de définitions). Structuellement, une liste de définitions est composée de trois éléments (balises) imbriquées, <dl>, <dt> et <dd> définis comme suit :

Balises	Description
<b>dl</b>	definition list = liste de définitions
<b>dt</b>	definition list term = un terme de la liste de définitions
<b>dd</b>	definition list definition = une définition de la liste de définitions (associée à un terme)

#### Exemple

<p>Voici une liste de termes et de leurs définitions :</p>

```

<dl>
  <dt>Développeur</dt>
  <dd>Un développeur est un informaticien qui programme des logiciels.</dd>
  <dt>Web</dt>
  <dd>Réseau maillé de serveurs d'informations formant une toile
  d'araignée.</dd>
</dl>

```

## 7. Compléments de balises

### 7.1. Insertion d'images

#### a. insertion simple

En XHTML, insérer une image revient à placer un lien vers le fichier qui la contient, avec la balise `<img ... />`. Elle contient obligatoirement l'attribut **src** qui indique le nom du fichier image à afficher.

#### Exemple

```

```

L'attribut **alt** d'une balise image permet d'afficher un texte de remplacement pour ceux qui naviguent en mode texte, ou pour les personnes handicapées ou en cas d'absence du fichier image. C'est aussi un moyen de renseigner les moteurs de recherche.

L'attribut **title** permet d'afficher un texte dans une bulle au passage de la souris sur l'image.

Les attributs **width** et **height** permettent de préciser les dimensions, en pixel, que doit prendre l'image (largeur et hauteur). Le cas le plus simple consiste à ne définir qu'un seul de ces attributs, car l'autre sera calculé automatiquement pour que les proportions soient respectées.

#### b. Insertion d'image comme une étiquette de lien

L'utilisation d'une image comme étiquette dans un lien ne pose bien entendu aucun problème. Il suffit de placer une balise `<img ... />` dans une balise `<a href...>...</a>`.

#### Exemple

```
<a href="Ma première page.html">  </a>
```

Il faut noter que par défaut les images cliquables ont une bordure de la couleur des liens. Pour la supprimer, il faut utiliser l'attribut **border** (en lui attribuant la valeur 0).

### 7.2. Insertion des animations, des sons et des vidéos

La balise `<object ...> ... </object>` permet d'insérer des objets tels que le son, la vidéo, l'animation, etc. ou les documents de type html, pdf, etc.

Il est possible de spécifier des informations au sein de la balise **<object>** pour paramétrer le contenu en utilisant la balise **<param .../>**.

### a. Insertion d'une Animation Flash

L'insertion d'une animation Flash se fait comme dans l'exemple suivant (insertion d'une animation Flash au format swf, **Animation.swf**):

```
<object type="application/x-shockwave-flash" data=" Animation.swf"
  width="150" height="100">
  <param name="src" value=" Animation.swf" />
  <!--Texte alternatif -->
  Ceci est une animation Animation.swf
</object>
```

La balise **<object ...>** possède plusieurs attributs,

L'attribut **data**, tout comme l'attribut **value** de la balise **<param>**, indique la source de la vidéo (le nom du fichier Shockwave Flash en .swf, format utilisé généralement par les hébergeurs de vidéo). Ces deux attributs devront donc avoir la même valeur. Bien sûr, sans cette ligne, la vidéo ne fonctionnerait pas.

**width** et **height** pour donner la largeur et la hauteur de cette animation sur l'écran. Le nom du fichier est répété dans une balise paramètre **<param ... />**.

Le texte alternatif contient un texte, pour les personnes handicapées ainsi que pour tous les navigateurs qui ne pourront pas afficher cette animation.

### b. Insertion d'une Vidéo

Il existe plusieurs techniques et plusieurs formats pour insérer une vidéo sur une page web. Dans chaque cas, il suffit de posséder sur l'ordinateur le logiciel capable de lire le format vidéo choisi.

#### b.1. Insertion d'une vidéo Flash

Une méthode courante et assez universelle consiste à insérer la vidéo en tant qu'objet Flash, comme sur les sites de partage de vidéos Youtube et Dailymotion. Il suffira alors que FlashPlayer soit installé sur l'ordinateur.

Le code XHTML utilise donc la syntaxe précédente et nécessite la présence de deux fichiers:

- Le fichier vidéo converti au format flv (Flash Vidéo).
- Un lecteur vidéo Flash, en quelque sorte une télévision qui s'affichera sur la page et qui diffusera le fichier vidéo (player\_flv\_multi.swf, flvplayer.swf, ...).

L'exemple suivant permet d'insérer une séquence vidéo flash ;

```
<object type="application/x-shockwave-flash" data="flvplayer.swf"
width="300" height="300">
  <param name="movie" value="flvplayer.swf" />
  <param name="flashvars" value="Video.flv" />
  <!--Texte alternatif -->
  C'est une séquence Flash vidéo
</object>
```

D'autres paramètres peuvent être ajoutés comme : par exemple celui qui autorise la fonction

**<param name="allowFullScreen" value="true" />** : Autorise la fonction plein écran

**<param name="flashvars" value="autostart=true&file= Video.flv" />** : programme un démarrage automatique de la vidéo avec l'affichage de la page,

Il faut mentionner qu'il est possible de visualiser une séquence Flash distante en écrivant son adresse web (URL) dans le paramètre **Value**.

## b.2. Insertion d'une vidéo de type AVI

Une deuxième manière d'insérer une vidéo est d'utiliser les types de vidéo comme le type AVI. L'exemple suivant représente une insertion d'un fichier Video.avi dans une page Web :

```
<object type="video/x-msvideo" data="Video.avi" width="320" height="255">
  <param name="src" value="Video.avi" />
  <param name="autoStart" value="0" />
  <!--Texte alternatif -->
  C'est une séquence Avi
</object>
```

## c. Insertion d'un fichier son

De même que pour l'insertion des fichiers vidéo, Il existe plusieurs techniques et plusieurs formats pour insérer un fichier audio sur une page web. Dans chaque cas, il suffit de posséder sur l'ordinateur le logiciel capable de lire le format audio choisi. L'exemple suivant représente une insertion d'un fichier audio.mp3 dans une page Web :

```
<object type="audio/mpeg" data="son.mp3" width="400px" height="50px">
  <param name="src" value="son.mp3"/>
  <param name="autoplay" value="true"/>
</object>
```

## Remarque

Dans les balises déjà vues, certaines comme `<p>... </p>` provoquent un retour à la ligne, alors que d'autres comme `<em>...</em>` laissent le texte concerné à sa place, c.à.d à la suite des mots précédents. La première catégorie est dite éléments en bloc (*in bloc*), la deuxième, quant à elle, est dite en ligne (*in ligne*).

## 8. Les frames (cadres)

Les frames (balise `<frameset ...> ...</frameset>`) permettent d'obtenir une ou plusieurs divisions horizontales et/ou verticales de la fenêtre du navigateur, et ainsi de disposer "virtuellement" de plusieurs fenêtres. Pour créer une page contenant des frames il faut :

- Définir le découpage (frameset).
- Définir le contenu des cadres (pages HTML classiques en général).

### Exemple

Division de la fenêtre principale en deux colonnes dont la deuxième est divisée en deux lignes :



Le code html qui permet de générer cette fenêtre est le suivant (frames imbriquées):

```
<frameset cols="300,*" border="0" frameborder="0" framespacing="0">
  <frame name="Sommaire" src="sommaire.html" >
  <frameset rows="*,450">
    <frame name="Description" src="description.html">
    <frame name="Image" src="image.html">
  </frameset>
</frameset>
```

Comme toute autre balise de XHTML, la balise `<frameset ...>` possède plusieurs attributs :

**rows|cols** : permet de définir le nombre de colonnes et/ou le nombre de lignes, \* est utilisé pour assigné le reste de la répartition à la colonne ou la ligne restante.

**<frame...>** : est une balise, elle permet la définition des frames, elle possède les propriétés suivantes :

**name** : le nom de la frame

**src** : le fichier cible à afficher

**scrolling** : active ou désactive l'ascenseur

**border** : permet de spécifier l'épaisseur de la bordure.

**frameborder** : permet de spécifier l'apparition ou non de la bordure.

**framespacing** : permet de spécifier l'espacement entre les frames.

Les trois dernières propriétés peuvent être incluses au début dans la balise **<frameset ...>**.

### Remarque

Il faut noter que les frames sont de moins en moins utilisés au profit des technologies CSS et back-end à cause des défauts qu'ils présentent tels que :

Dénaturation radicale du document Web ;

Rendent l'indexation délicate ;

Posent des problèmes d'accessibilité aux personnes handicapées (non-voyants notamment) ;

etc...

## 9. Les formulaires

Les formulaires sont le moyen le plus utilisé pour communiquer (saisir des données) entre une application et un utilisateur. En XHTML, la balise **<form> ... </form>** est le moyen de déclaration d'une zone de formulaire dans laquelle les utilisateurs peuvent saisir des données qui pourront être, par la suite, récupérées et traitées par le serveur de la page ou du site web.

### Exemple

#### Questionnaire d'identité :

Nom :  Prénom :

Date de naissance :  Lieu de naissance :

Adresse :

Cliquez sur  pour soumettre vos données, sinon

Cet exemple représente le rendu du code suivant :

```
<form method="post" action="traitement">
```

```
  <h2>Questionnaire d'identité :</h2>
```

```
  <p> Nom : <input type = "text" name="nom" size="30">
```

```
  Prénom : <input type = "text" name="prenom" size="30"></p>
```

```
  <p> Date de naissance : <input type = "text" name="datnais" size= "20">
```

```
  Lieu de naissance : <input type = "text" name="lieunais" size="30"> </p>
```

```
  <p> Adresse : <input type = "text" name="adresse" size="60"> </p>
```

```
  <p> Cliquez sur <input type="submit" value="Valider"> pour soumettre votre données,
```



```

    sinon <input type="reset" value="Annuler"></p>
</form>

```

### Remarque

La mise en forme du formulaire ne sera pas abordée dans ce qui suit, puisque c'est le rôle des feuilles de style CSS. En ce qui concerne l'envoi et les traitements des données saisies, ça dépasse le cadre de l'XHTML. Il faut un langage côté serveur, comme le PHP par exemple.

## 8.1. Syntaxe générale

La syntaxe générale de la balise `<form> ... </form>` est la suivante :

```

<form method="get/post" action="traitement" >
    <!--élément du formulaire -->
    ...
    <input type="submit" value ...>
    <input type="reset" value ...>
</form>

```

L'attribut **method** sert à définir la méthode d'envoi (**get** ou **post**) des données. **get** envoie les données par URL, c'est-à-dire que les données saisies par l'utilisateur dans le formulaire sont renvoyées dans l'URL. Ici, la longueur des données envoyées est limitée à 255 caractères. Avec **post** il n'y a pas de limite, on peut envoyer la quantité de données qu'on souhaite à condition de respecter les limitations du serveur.

L'attribut **action** définit la destination où sont envoyées les données ou la méthode avec laquelle elles vont être traitées.

Pour qu'un formulaire puisse être traité il faut qu'il soit soumis à l'aide de la balise `<input type="submit" value .../>`. Pour annuler la saisie il faut utiliser la balise `<input type="reset" value ...>`

### Remarque

1. Pour envoyer les données d'un formulaire dans un email il suffit de mentionner ce dernier dans l'attribut **action** précédé de la mention «mailto» (exemple : `method="post" action="mailto:adressemail@yahoo.fr" enctype="text/plain">`).

2. Une autre méthode de soumission des données d'un formulaire est l'utilisation de javascript en appliquant la fonction "submit()" au formulaire (en se basant sur son identifiant).

3. `<input type="image" name="Valider" type="submit" src="valider.png">`

(l'image valider.png doit être dans le même dossier que la page).

## 8.2. Autres éléments utilisés avec la balise `<input type=...>` dans un formulaire

### a. Zone de saisie simple

```
<input type="text" name="nom zone saisie" value="texte par défaut" size="longueur zone
saisie" maxlength="longueur max saisie">
```

### b. Case à cocher «checkbox »

```
<input type="checkbox" name="nom case" value="valeur case (0,1)" >
```

#### Exemple

```
choix 1 <input type="checkbox" name="case_1" value="1" checked> <br>
choix 2 <input type="checkbox" name="case_2" value="1"><br>
choix 3 <input type="checkbox" name="case_3" value="1"><br>
choix 4 <input type="checkbox" name="case_4" value="1">
```

Choix 1	<input checked="" type="checkbox"/>
Choix 2	<input type="checkbox"/>
Choix 3	<input type="checkbox"/>
Choix 4	<input type="checkbox"/>

#### Principes

Plusieurs cases peuvent avoir l'option checked.

Les quatre cases n'ont pas le même nom.

**value** indique la valeur qui sera retournée en fonction des cases cochées.

Toutes les cases peuvent être décochées.

### c. Bouton radio bouton « radiobutton »

```
<input type="radio" name="nom radio" value="valeur radio (0,...)" >
```

#### Exemple

```
choix 1 <input type="radio" name="boutonradio" value="1" checked> <br>
choix 2 <input type="radio" name="boutonradio" value="2"><br>
choix 3 <input type="radio" name="boutonradio" value="3"><br>
choix 4 <input type="radio" name="boutonradio" value="4">
```

Choix 1	<input checked="" type="radio"/>
Choix 2	<input type="radio"/>
Choix 3	<input type="radio"/>
Choix 4	<input type="radio"/>

#### Principes

Une seule case peut avoir l'option checked

Les 4 cases ont le même nom

value indique la valeur qui sera retournée en fonction de la case cochée.

## 8.3. Liste de choix

```
<select name="nom de la liste" size="1" < >1">
  <option ... >texte de la ligne 1 </option>
  ...
</select>
```

Lorsque `size="1"` la liste prend la forme d'une liste déroulante sinon elle prend la forme d'une liste défilante.

### Exemple

```
<select name="Liste de choix" size="1" >
  <option > Choix 1 </option>
  <option > Choix 2 </option>
  <option > Choix 3 </option>
</select>
```

### Principes

Une seule option est sélectionnée (`selected`) elle sera retournée dans le nom de la sélection.

On peut présélectionner l'élément affiché dans la boîte d'entrée (par défaut, le premier élément de la liste sera retenu). On utilise pour ce faire l'attribut `selected` de la balise `<option>`. Pour désactiver un élément on utilise l'attribut `disabled`.

### 8.4. Zone de saisie multilignes

```
<textarea name="nom de la zone" rows="r" cols="c"> ... </textarea>
```

Cette balise permet de définir une zone de texte pour saisir des données de taille plus importante que dans une case de texte. `rows` représente la taille de la zone de saisie en nombre de ligne, `cols` représente la taille de la zone de saisie en nombre de colonnes.

### Exemple

```
Commentaires <textarea name="commentaires" rows="4" cols="5"> faites vos
commentaires ici </textarea>
```