

وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR ANNABA  
FACULTE DE TECHNOLOGIE  
DEPARTEMENT D'INFORMATIQUE



جامعة باجي مختار عنابة  
كلية التكنولوجيا  
قسم الاعلام الالي

Domaine : Mathématique-informatique  
Filière : Informatique  
Spécialité : Master Gestion et Analyse des Données  
Massives (G.A.D.M)

**Polycopié du support du cours intitulé**

## **Outils de l'internet I**

**Préparé par : Dr. Mohamed Amine YAKOUBI**  
email : yakoubi\_amine@hotmail.fr

# Table des matières

Liste des figures.....	iii
Liste des Tableaux.....	iv
Avant-Propos.....	v
Chapitre 1.....	1
1 Qu'est-ce qu'un réseau ?.....	2
2 Qu'est-ce que internet ?.....	2
3 Historique d'internet.....	3
4 Les services d'internet.....	4
Chapitre 2.....	5
1 Web ?.....	6
2 HTML (Hyper Text Markup Language) ?.....	6
3 Exercices.....	16
Chapitre 3.....	20
1 XML, C'est quoi ?.....	21
2 XML et HTML.....	21
3 Positionnement de XML par rapport à SGML.....	22
4 Principe et apport.....	22
5 Syntaxe et structure.....	22
6 Parseurs XML.....	37
7 Editeurs XML.....	38
8 Exercices.....	38
Chapitre 4.....	40
1 Introduction.....	41
2 Courrier électronique ?.....	41
3 Fonctionnement d'un email ?.....	41
4 Création d'un compte électronique.....	41
5 Adresse électronique.....	42
6 Messagerie électronique.....	42
7 Serveurs de messagerie.....	42
8 Protocoles de communication.....	43
9 Conclusion.....	43
10 Exercices.....	43
6. Références.....	45

## Liste des figures

<b>Figure 1:</b> Réseau Informatique .....	2
<b>Figure 2:</b> Internet .....	2
<b>Figure 3:</b> Fonctionnement du web .....	6
<b>Figure 4:</b> Structure d'un document HTML .....	8
<b>Figure 5:</b> Tableaux en HTML .....	9
<b>Figure 6:</b> Exemple d'un tableau en HTML .....	10
<b>Figure 7:</b> Nombre et taille des colonnes d'un tableau .....	10
<b>Figure 8:</b> Les images (map) en HTML .....	12
<b>Figure 9:</b> Frames .....	12
<b>Figure 10:</b> Exemple frames .....	13
<b>Figure 11:</b> Champs Texte simple .....	14
<b>Figure 12:</b> Champs mot de passe .....	14
<b>Figure 13:</b> Zone du texte .....	15
<b>Figure 14:</b> Champs de selection .....	15
<b>Figure 15:</b> Case à cocher .....	15
<b>Figure 16:</b> bouton radio .....	16
<b>Figure 17:</b> Bouton d'envoi .....	16
<b>Figure 18:</b> Boutin d'annulation .....	16
<b>Figure 19:</b> Bouton d'envoi d'un fichier .....	16
<b>Figure 20:</b> Origines de XML .....	21
<b>Figure 21:</b> Exemple en XML .....	23
<b>Figure 22:</b> Exemple DTD .....	23
<b>Figure 23:</b> Prologue .....	24
<b>Figure 24:</b> DTD externe .....	25
<b>Figure 25:</b> DTD interne .....	25
<b>Figure 26:</b> Les éléments .....	25
<b>Figure 27:</b> Eléments vides .....	25
<b>Figure 28:</b> Exemple .....	26
<b>Figure 29:</b> Attributs .....	26
<b>Figure 30:</b> CDATA .....	26
<b>Figure 31:</b> Exemple .....	26
<b>Figure 32:</b> Commentaire .....	27

## Liste des Tableaux

<b>Tableau 1:</b> Caractères spéciaux.....	7
<b>Tableau 2:</b> Entités prédéfinies.....	27
<b>Tableau 3:</b> Elément vs Attributs.....	27
<b>Tableau 4:</b> ELEMENT.....	28
<b>Tableau 5:</b> Séparateur.....	29
<b>Tableau 6:</b> Indicateurs.....	29
<b>Tableau 7:</b> Attributs.....	31
<b>Tableau 8:</b> Déclaration par défaut.....	31

## **Avant-Propos**

Ce polycopié est rédigé pour servir comme support pédagogique à l'intention des étudiants inscrits en première année Master de la filière Informatique, spécialité Gestion et Analyse des Données Massives (G.A.D.M). Il peut également être utile pour d'autres filières où une résolution des problèmes par les outils de développement web s'impose.

Le contenu de ce support, correspond au programme de la matière "Outils de l'internet I". Il met ainsi l'accent sur les langages de programmation et plus particulièrement sur les langages de développement web. Il permet donc l'acquisition des connaissances sur la manipulation de ces langages. En effet, il suppose la connaissance de notions sur la conception et la réalisation des pages web.

L'objectif de ce support du cours est de se familiariser avec les langages HTML, XML, XHTML et les protocoles des courriers électroniques. Une explication de ce que c'est qu'Internet, les services offerts par ce réseau ainsi qu'un survol historique sont présentés au début de ce support de cours.

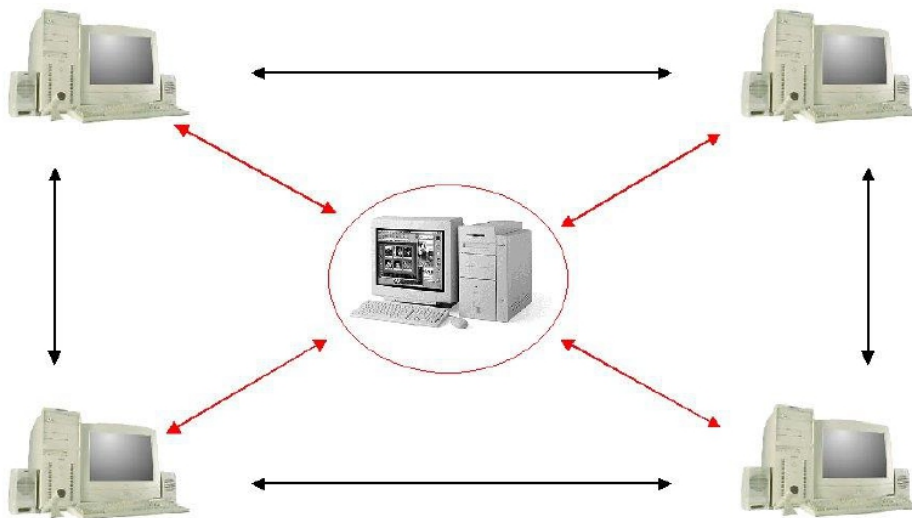
Sur le plan pratique l'étudiant doit résoudre des exercices à la fin de chaque chapitre.

# Chapitre 1

## Internet

## 1 Qu'est-ce qu'un réseau ?

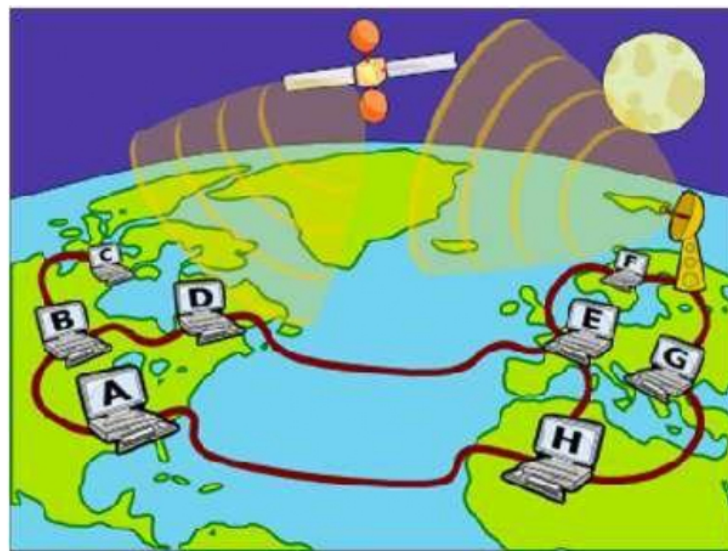
Un réseau est constitué d'ordinateurs reliés entre eux à l'aide des moyens de communications actuels. **Figure 1**



*Figure 1: Réseau Informatique*

## 2 Qu'est-ce que internet ?

Internet (INTERconnexion NETwork) est un ensemble des ordinateurs du monde entier reliés entre eux (réseau planétaire) pour échanger des informations. Les ordinateurs de l'Internet sont reliés par des câbles qui traversent les pays et les océans ou par des satellites. **Figure 2**



*Figure 2: Internet*

Internet fonctionne sur la base de protocoles dont le plus connu est TCP-IP. Ce protocole, en donnant une identité à chaque ordinateur (une adresse électronique), organise la relation entre les différentes machines sur le Net.

L'internet est un univers où chacun peut produire ou utiliser, donner de l'information ou en trouver, et participer librement à cette planète de l'information.

L'internet ne dépend actuellement d'aucun organisme de normalisation. Ses produits naissent et durent ou disparaissent selon un seul critère, leur appréciation par les utilisateurs.

Pour le grand public et les médias internet et WWW (World Wide Web) ne font qu'un. En fait, le WWW est un des services les plus utilisés sur internet. Il est constitué de sites, construits par des particuliers ou des entreprises, abritant des pages constituées de textes, images, sons, animations, etc. Ces pages peuvent être consultées grâce à des logiciels particuliers, dit de consultation de pages web : IE, NetScape, ...

### **3 Historique d'internet**

Au début des années 60, le ministère de la Défense américain travaille sur l'idée de relier entre eux les ordinateurs militaires pour partager les informations. C'est la naissance d'ARPANET. Les chercheurs mettent au point la transmission par paquets, c'est à-dire que le message est découpé en plusieurs morceaux envoyés séparément et reconstitués à l'arrivée.

**1969** : il devient possible de connecter entre elles des machines de conceptions différentes.

**1972** : ARPANET connecte 23 centres.

**1974** : Vinton Cerf et Bob Kahn publient un rapport décrivant l'architecture des protocoles qui permettent la circulation des informations. C'est la naissance du sigle TCP/IP.

**1977** : ARPANET connecte 111 centres.

**1978** : naissance d'un nouveau réseau appelé USENET qui regroupe les utilisateurs d'ordinateurs utilisant UNIX comme langage informatique.

**Années 80** : multiplication des réseaux informatiques universitaires.

**1984** : ARPANET n'est plus un réseau militaire.

**1986** : l'Internet regroupe 400 réseaux.

Jusqu'alors le réseau sert à faire transiter des informations.

**1990** : au C.E.R.N., le Centre européen de recherches nucléaires, TimBerners-Lee développe le premier système de documentation fonctionnant en « hypertexte » et met au point le premier logiciel (browser – navigateur) permettant d'utiliser ce langage. C'est la naissance du World Wide Web, d'abord utilisé par les physiciens.

**1993** : il y a plus de 200 sites web.

**1994** : apparition des premiers fournisseurs d'accès privés en France.

**1995** : 20 millions d'utilisateurs.

**2002** : ± 36 millions de sites web. 600 millions d'utilisateurs.

**2020** : 4,79 milliards de personnes se sont connectées à internet en 2020

#### **4 Les services d'internet**

Internet offre à ses utilisateurs une très grande variété de services, parmi ceux-ci, trois fonctions de base :

- ❖ **le courrier** : le service de messagerie électronique permet d'échanger des messages avec des millions de personnes à travers le monde.
- ❖ **le transfert de fichiers** : Internet est utilisée comme une grande bibliothèque distribuée à travers la planète. Elle contient des fichiers de tout type et en particulier des logiciels mis à la disposition de tous, où qu'elles soient.
- ❖ **la connexion à distance** : la connexion à des ordinateurs éloignés est largement utilisée dans le monde scientifique, les commandes rlogin (remote login) ou telnet existent depuis le début réseau et permettent de travailler en mode réparti. Il donnent la possibilité d'utiliser des ressources offertes sur un ordinateur éloigné, par exemple les bases de données, la puissance de calcul, etc.

# Chapitre 2

## HTML (Hyper Text Markup Language)

## 1 Web ?

### 1.1 Qu'est-ce qu'un web ?

Web, est un des services disponibles sur Internet. D'autres services disponibles sur Internet sont par exemple la messagerie électronique (email) ou le transfert de fichier (FTP).

Le Web ("la toile" en français) est un système permettant de visualiser et d'échanger des informations à distance. Les informations sont présentées sur des pages Web et reliées les unes aux autres par des liens hypertextes.

### 1.2 Qu'est-ce qu'une page web et un site web ?

Une page Web est généralement un fichier texte écrit dans un langage informatique baptisé HTML. Ce fichier peut contenir des images ou d'autres contenus multimédias (audio, vidéo, application).

Un site Web est un ensemble de pages Web reliées les unes aux autres par des liens hypertextes et accessible en ligne à une adresse.

### 1.3 Fonctionnement du web ?

Le web se fonctionne sous une architecture appelée *client/serveur*. Un serveur Web est un logiciel qui agit comme un récepteur de requêtes envoyées via un navigateur et répond ensuite à la requête sous la forme d'une page de site Web ou plus généralement dans un document HTML. Cependant, le serveur Web peut avoir deux significations différentes, à savoir en tant que partie du matériel et en tant que partie du logiciel. **Figure 3**

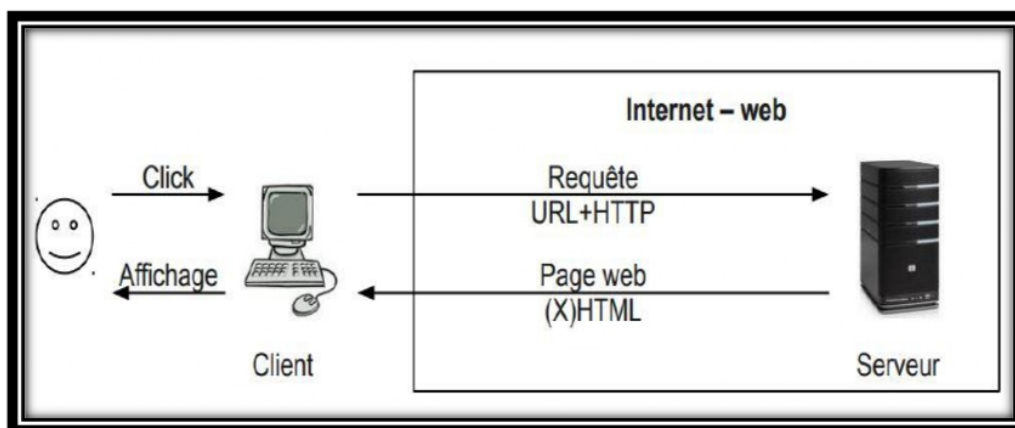


Figure 3: Fonctionnement du web

## 2 HTML (Hyper Text Markup Language) ?

### 2.1 Définition ?

C'est un langage de balisage créé au CERN, en 1989 par Berners-Lee. À l'origine, c'était pour une mise en forme de documents hypertextes (liens). Aujourd'hui, on peut l'utiliser pour la structuration ainsi que l'insertion d'objets (Flash, vidéo, Java, etc.). Le HTML est pour quasiment tous les sites web.

## 2.2 Contenu ?

### 2.2.1 Balise

C'est l'élément de base du langage. Elle a le **Format** suivant :

*<balise ouvrante> contenu <balise fermante>*

*Exemple texte en gras* : `<p> HTML : <b>Notes de cours </b> </p>`

Toutefois il y a une **exception** pour les balises vides. exemple : `<br>`

### 2.2.2 Attributs

C'est un élément, présent au sein de *la balise ouvrante*, permettant de définir des propriétés supplémentaires *clé = valeur*. Ils permettent de décrire les propriétés d'une balise.

*Exemple Alignement à droite* : `<p align="right"> Bonjour </p>`

### 2.2.3 Commentaires

Ils permettent de *décrire* le code. Ils ne sont pas visibles par le navigateur.

`<!-- Commentaire -->`

*Exemple commentaire* : `<!-- Je suis un commentaire-->`

### 2.2.4 Caractères spéciaux

Le HTML demande de respecter un codage spécifique pour *les caractères spéciaux*. Ainsi, tous les caractères accentués doivent être « traduits » dans des suites de caractères qui débutent par & et se terminent par un point-virgule (;).

*Exemple* une liste des caractères spéciaux (**Tableau 1**) :

*Tableau 1: Caractères spéciaux*

Caractères spéciaux	Entités HTML	Caractère ASCII	Description caractères
	&nbsp;	&#160;	Espace insécable
'	&apos;	&#39;	Guillemet simple droit
...	&hellip;	&#8230;	Points de suspension
©	&copy;	&#169;	Copyright
&	&amp;	&#38;	Et commercial

## 2.3 Structure ?

La structure de base d'un document HTML se compose de 5 éléments (**Figure 4**).

*Remarque* Chaque document Web doit inclure une et une seule instance de *DOCTYPE*, `<html>`, `<head>`, `<body>` et `<title>`.

`<!DOCTYPE type du document>` **Type du document**

<code>&lt;html&gt;</code>	<b>Elément html</b>
<code>&lt;head&gt;</code>	
En-tête du document	<b>En-tête du document</b>
<code>&lt;/head&gt;</code>	
<code>&lt;title&gt;</code>	
Titre du document	<b>Titre du document</b>
<code>&lt;/title&gt;</code>	
<code>&lt;body&gt;</code> Ou <code>&lt;frameset&gt;</code>	
.....	
<code>&lt;/body&gt;</code> Ou <code>&lt;/frameset&gt;</code>	<b>Corps du document</b>
<code>&lt;/html&gt;</code>	

Figure 4: Structure d'un document HTML

### 2.3.1 Type du document

Une déclaration DOCTYPE doit être précisée sur la première ligne de chaque document web. Le DOCTYPE indique aussi au navigateur Web dans quelle version de HTML la page est écrite. Il existe trois types de document html :

2.3.1.1 Strict fait l'impasse sur les caractéristiques obsolètes des versions précédentes

```

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

2.3.1.2 Transitional compatibilité avec les versions précédentes

```

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

2.3.1.3 Frameset définit des pages avec cadres

```

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

### 2.3.2 Elément HTML

Immédiatement après la déclaration DOCTYPE se trouve l'élément `<html>`. L'élément `<html>` indique au navigateur que la page sera formatée en HTML et, éventuellement, dans quelle langue mondiale se trouve le contenu de la page.

### 2.3.3 Entête

L'élément `<head>` entoure tous les éléments spéciaux "en coulisses" d'un document Web. La plupart de ces éléments ne s'affichent pas directement sur la page Web. Elle est destinée au navigateur : titre du document, mots clés, css, ...etc.

### 2.3.4 Titre

L'élément `<title>` définit le texte qui s'affichera dans la barre de titre du navigateur Web.

### 2.3.5 Corps

L'élément `<body>` entoure tout le contenu réel (texte, images, vidéos, liens, etc.) qui sera affiché sur notre page Web. Détermine le contenu du document. On peut utiliser des balises de mise en forme (gras, italique, ...etc) aussi.

### 2.4 Lien ?

Il est utilisé pour créer un lien cliquable vers un autre emplacement. Il existe quatre types de liens :

#### 2.4.5 URL (Uniforme Resource Locator)

Une URL est simplement l'adresse d'une ressource donnée, unique sur le Web.

```
<a href="http://www.univ-annaba.org">Université B.M.A</a>
```

#### 2.4.6 Lien absolu

Il conduira le visiteur vers un site Web externe à notre propre site.

```
<a href="http://www.univ-annaba.org" target="_blank">Université B.M.A</a>
```

#### 2.4.7 Lien relatif

La navigation à l'intérieur d'un même site. On utilise le même format mais tape simplement le nom du fichier directement dans l'attribut href :

```
<a href="page2.html" > Page2 </a>
```

#### 2.4.8 Lien relatif

Permet de référencer un endroit précis dans un document.

```
<a href=" #contact" > contact </a>
```

### 2.5 Tableaux ?

Les tableaux nous permettent d'afficher des informations de manière prévisible et uniforme sur une page Web. Les tableaux sont bien adaptés pour présenter des données conformes aux lignes et aux colonnes.

Le contenu des cellules de tableau est facile à formater de manière cohérente.

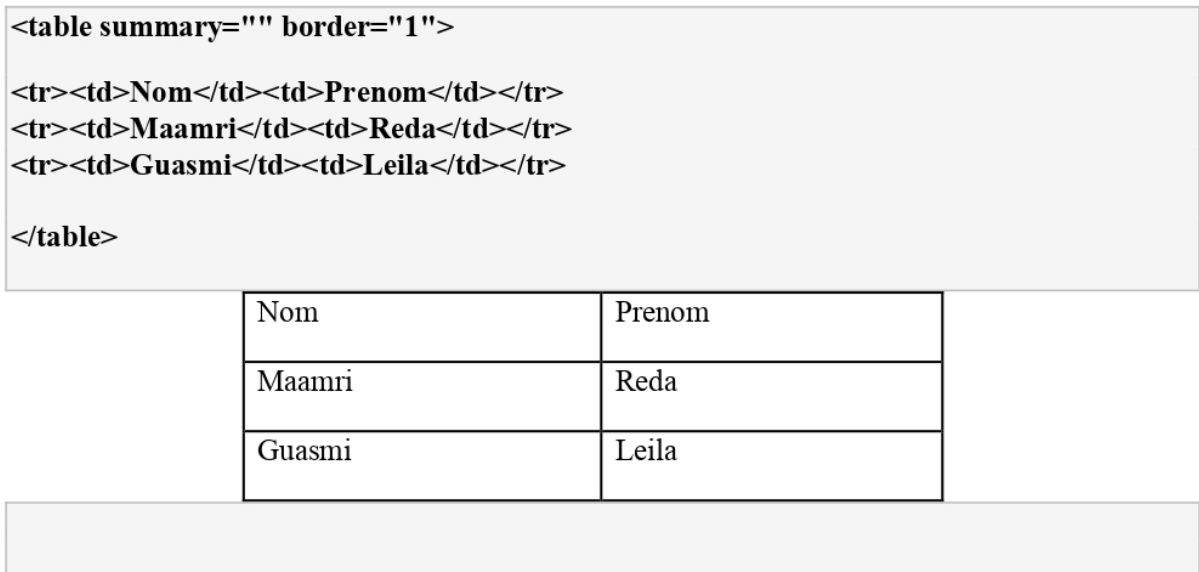
#### 2.5.1 Lignes et colonnes

Le tableau est composé d'un nombre de lignes et de colonnes. Les balises html sont (Figure 5):

<code>&lt;table&gt;</code>	définition du tableau
<code>&lt;tr&gt;</code>	définition d'une ligne
<code>&lt;td&gt;</code>	définition d'une cellule

Figure 5: Tableaux en HTML

*Exemple* : Dans ce code, nous avons créé un tableau et défini trois lignes de tableau, chacune avec deux cellules de données de tableau (c'est-à-dire deux colonnes) **Figure 6**.



*Figure 6: Exemple d'un tableau en HTML*

### 2.5.2 D'autres balises

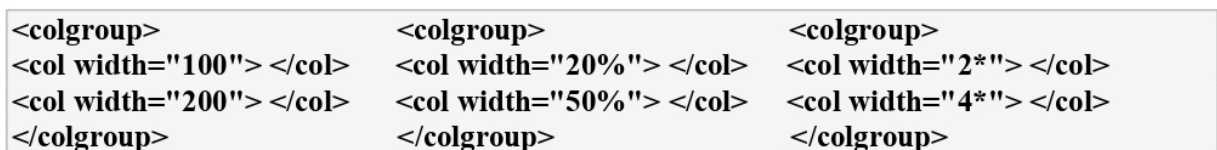
- **<caption>** : définition d'un titre de tableau
  - attribut « **align** » Peut être positionné au dessus ou au dessous du tableau.
  - **<th>** : définition d'une cellule d'en-tête. La cellule apparaît avec une mise en forme différente
  - **<thead>**, **<tbody>** et **<tfoot>** : découpe logique du tableau en en-tête, corps et pied

Utile lorsque des règles de quadrillage sont spécifiées :

- ✓ **border** : épaisseur des bordures du tableau (0 = aucune bordure affichée)
- ✓ **cellpadding** : marge intérieure des cellules (entre le contenu et les bords de la cellule)
- ✓ **cellspacing** : marge entre cellules
- ✓ **width** : largeur du tableau

### 2.5.3 Nombre et taille des colonnes

- **<colgroup>** et **<col>** : pré-définition des colonnes du tableau (nombre et taille)
- La taille est exprimée en : Pixels, Pourcentage, Unité de grandeur relative entre colonnes.



*Figure 7: Nombre et taille des colonnes d'un tableau*

## 2.6 Images ?

Les formats supportés sont : *GIF, JPEG, PNG*.

### 2.6.1 Les balises

La balise utilisée est :

```

```

Les attributs possibles sont :

- **src** : URL de la ressource image
- **alt** : description (s'affiche en « tooltip »)
- **width** : largeur de l'image affichée
- **height** : hauteur de l'image affichée Exemple :

Exemple :

- ``
- ``
- ``

### 2.6.2 Images et liens

Avec la balise **map** il est possible d'avoir une image avec des zones cliquable. Chacune de ces zones peut nous conduire à un lien.

Les zones peuvent être définies par la balise **area** dont les attributs possibles sont :

- **href** : URL de la ressource cible
- **shape** : forme de la zone (rect, circle, poly, default)
- **coords** : coordonnées de la zone
- **alt** : texte descriptif du lien

Exemple : **Figure 8**

```

<map name="panneaux" id="panneaux">
  <area shape="rect" coords="92,19,261,66" href="page-contact.html" alt="Contact" />
  <area shape="circle" coords="314,105,44" href="page-faq.html" alt="FAQ" />
  <area shape="poly" coords="18,100,42,83,190,103,183,148,36,132" href="page-home.html" alt="Accueil" />
</map>
```



Figure 8: Les images (map) en HTML

## 2.7 Frames ?

Les Frames sont en fait des cadres (frame=cadre en anglais) qui permettent de diviser votre page Web en plusieurs parties indépendantes l'une de l'autre mais pouvant également agir l'une sur l'autre... Ces différentes parties sont en fait de simples pages HTML ! ()

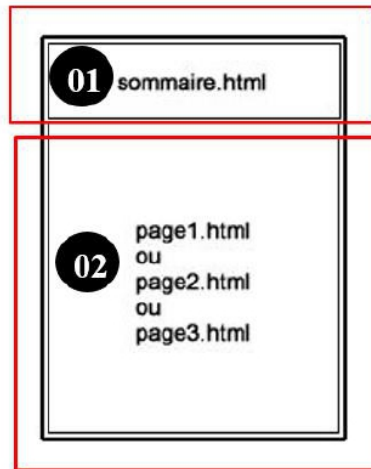


Figure 9: Frames

### 2.7.1 Balises

Découper une fenêtre en plusieurs zones, chacune peut afficher un contenu différents. (Figure 10)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HTML>
<HEAD>
<TITLE>A frameset document</TITLE>
</HEAD>
<FRAMESET rows="20%,80%">
  <FRAME name="fixed" src="init_fixed.html">
  <FRAME name="dynamic" src="init_dynamic.html">
```

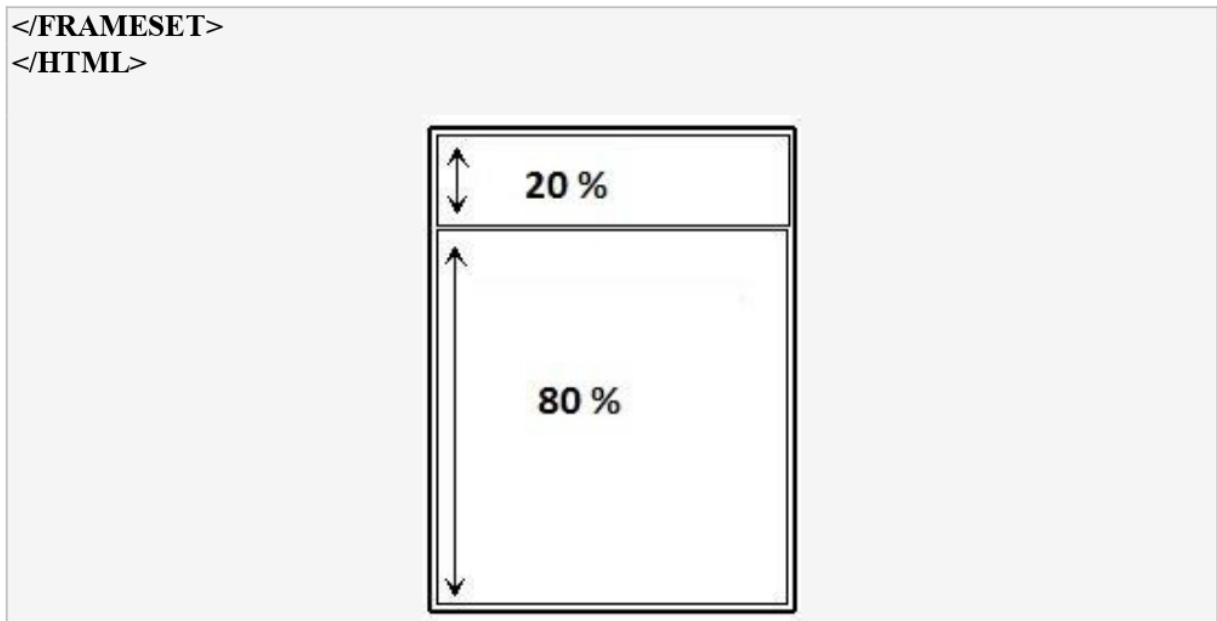


Figure 10: Exemple frames

### 2.7.2 Attributs

- **marginwidth** précise la largeur des marges latérales (nombre de pixels)
- **marginheight** précise la hauteur des marges hautes et basses (nombre de pixels)
- **scrolling** précise si l'on doit afficher un ascenseur ou non — les valeurs possibles sont yes, no et auto (valeur par défaut)
- **noresize** Indique que la taille de la zone ne peut pas être modifiée par l'utilisateur.
- **frameborder** Indique si le cadre doit être séparé des autres cadres (1 pour oui —valeur par défaut, 0 pour non)

## 2.8 Formulaires ?

Permettre au visiteur de saisir des informations. Un formulaire HTML ou XHTML permet de saisir et de transmettre des informations.

Un formulaire est déclaré grâce à la balise **<form>** dont on précisera généralement 3 attributs :

```
<form action="action" method="GET">...</form>
```

- **name** : le nom que l'on souhaite donner au formulaire
- **action** : le chemin vers la page où les données du formulaire seront envoyées. Généralement, il s'agit d'une page dite "dynamique" (en .php ou .asp par exemple), c'est à dire une page qui peut interagir avec le serveur pour sauvegarder les données dans une base.
- **Method** : la méthode d'envoi des données qui est soit "POST", soit "GET"

### 2.8.1 Method GET et POST

- En **GET**, les données du formulaire seront transmises via l'URL de la page. On verra alors des variables s'ajouter à la fin de l'URL dans la barre d'adresse.

```
http://www.example.com/page-envoi.html?nom=Gandhi
```

- En **POST**, les données du formulaire seront transmises via l'entête de la page WEB. Les variables seront alors invisibles pour l'utilisateur.

```
http://www.example.com/page-envoi.html
```

### 2.8.2 Champs Texte simple

Utiliser la balise `<input />` en précisant l'attribut `type="text"`. L'attribut `name` permet de spécifier le nom de la variable qui sera envoyé (**Figure 11**). L'attribut `value` permet de spécifier une valeur par défaut, c'est à dire que le champs sera affiché pré-rempli.

```
Veillez saisir votre prénom :<br /><input type="text" name="prenom" value="" />
```



Figure 11: Champs Texte simple

### 2.8.3 Champs Mot de passe

Utiliser la balise `<input />` en précisant l'attribut `type="password"`. (**Figure 12**)

```
Veillez saisir votre mot de passe :<br /><input type="password" name="motde  
passe" value="Annaba" />
```



Figure 12: Champs mot de passe

### 2.8.4 zone de texte

Pour un champs texte sur plusieurs lignes.

- Utiliser les balises `<textarea>` et `</textarea>`. (**Figure 13**)
- L'attribut `name` permet de spécifier le nom de la variable qui sera envoyé.
- L'attribut `rows` permet de préciser le nombre de lignes de la zone de texte.
- Le texte placé entre les balises `<textarea>` et `</textarea>` permet de spécifier une valeur par défaut.
- L'attribut `cols` permet de préciser le nombre affiché prérempli.

```
Votre message :<br /><textarea name="le-  
message" rows="6" cols="40"> Vous pouvez saisir ici un  
message.</textarea>
```



Figure 13: Zone du texte

### 2.8.5 Champs de sélection

Les champs select permettent de proposer une liste déroulante de choix. Utiliser les balises `<select>` et `</select>`. Placer autant de couple `<option>` et `</option>` que nous avons de choix. (Figure 14)

```
Vous êtes :<br />
<select name="profil">
  <option value="parti">Un particulier</option>
  <option value="profe">Un professionnel</option>
  <option value="insti">Un institutionnel</option>
</select>
```

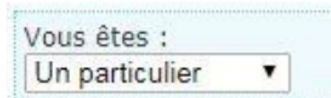


Figure 14: Champs de selection

### 2.8.6 Case à cocher

Utiliser la balise `<input />` en précisant l'attribut `type="checkbox"`. L'attribut `name` permet de spécifier le nom de la variable qui sera envoyé. L'attribut `value` permet de spécifier la valeur qui sera envoyée si la case est cochée. (Figure 15)

```
Quel type de prestation recherchez vous ?<br />
<input type="checkbox" name="interet" value="loc" /> Location de mobilier
<input type="checkbox" name="interet" value="achat" /> Achat de mobilier
```

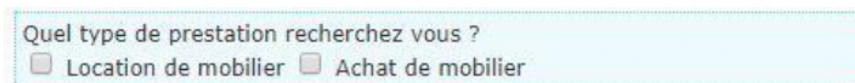


Figure 15: Case à cocher

### 2.8.7 Un bouton d'option ou «bouton radio»

Utiliser la balise `<input />` en précisant l'attribut `type="radio"`. (Figure 16)

```
Civilité :<br />
<input type="radio" name="civi" value="Mme" /> Madame
<input type="radio" name="civi" value="Mlle" /> Mademoiselle
<input type="radio" name="civi" value="Mr" /> Monsieur
```

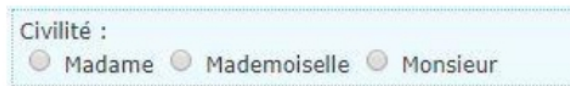


Figure 16: bouton radio

### 2.8.8 Bouton d'envoi

Utiliser la balise `<input/>` en précisant l'attribut `type="submit"`. L'attribut `value` permet de spécifier le texte qui sera affiché sur le bouton. Lors du clique sur ce bouton, les données du formulaires seront envoyées à la page précisée par l'attribut `action` du formulaire. (Figure 17)

```
<input type="submit" value="Envoyer" />
```



Figure 17: Bouton d'envoi

### 2.8.9 Bouton d'annulation ou «reset»

Utiliser la balise `<input />` en précisant l'attribut `type="reset"`. L'attribut `value` permet de spécifier le texte qui sera affiché sur le bouton. (Figure 18)

Lors du **clique** sur ce bouton, les données saisies par l'utilisateur **seront effacées**.

```
<input type="reset" value="Annuler" />
```



Figure 18: Boutin d'annulation

### 2.8.10 un bouton d'envoi d'un fichier «parcourir»

Utiliser la balise `<input type="file" />` Permet la sélection d'un fichier local (Figure 19)

`enctype="multipart/form-data"` doit être spécifié dans la balise `<form>`

```
<input type="file" name="fichier" />
```



Figure 19: Bouton d'envoi d'un fichier

## 3 Exercices

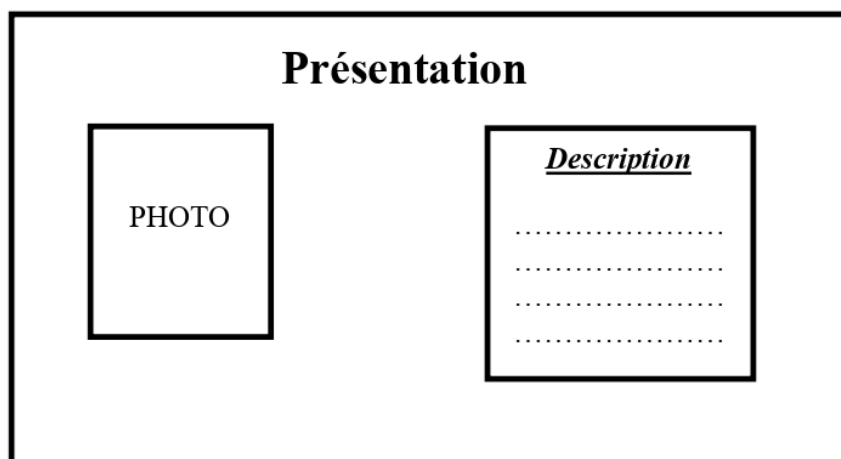
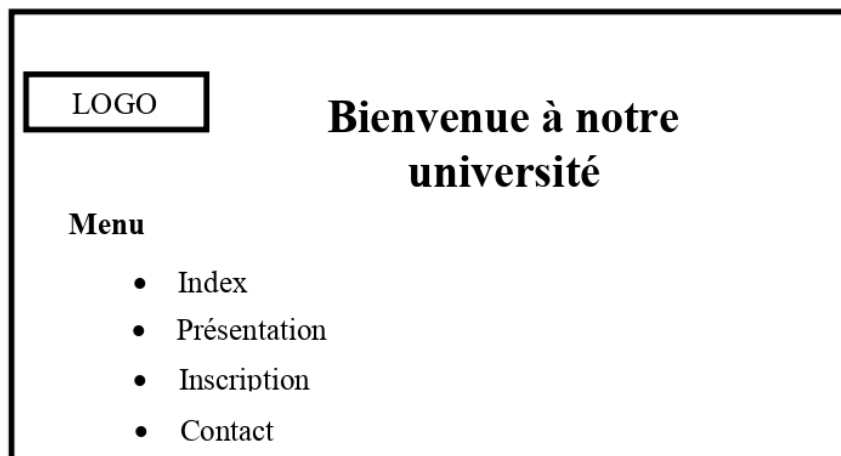
### 3.1 Exercice 1

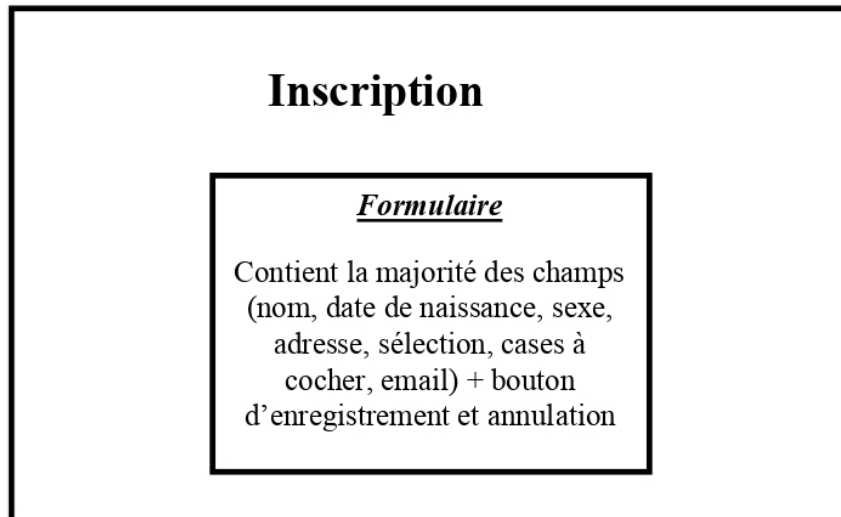
Créer un nouveau sous-dossier que vous nommerez "TP"

Vous créez 4 pages html que vous nommerez :

- index.html
- presentation.html
- Inscription.html
- contact.html

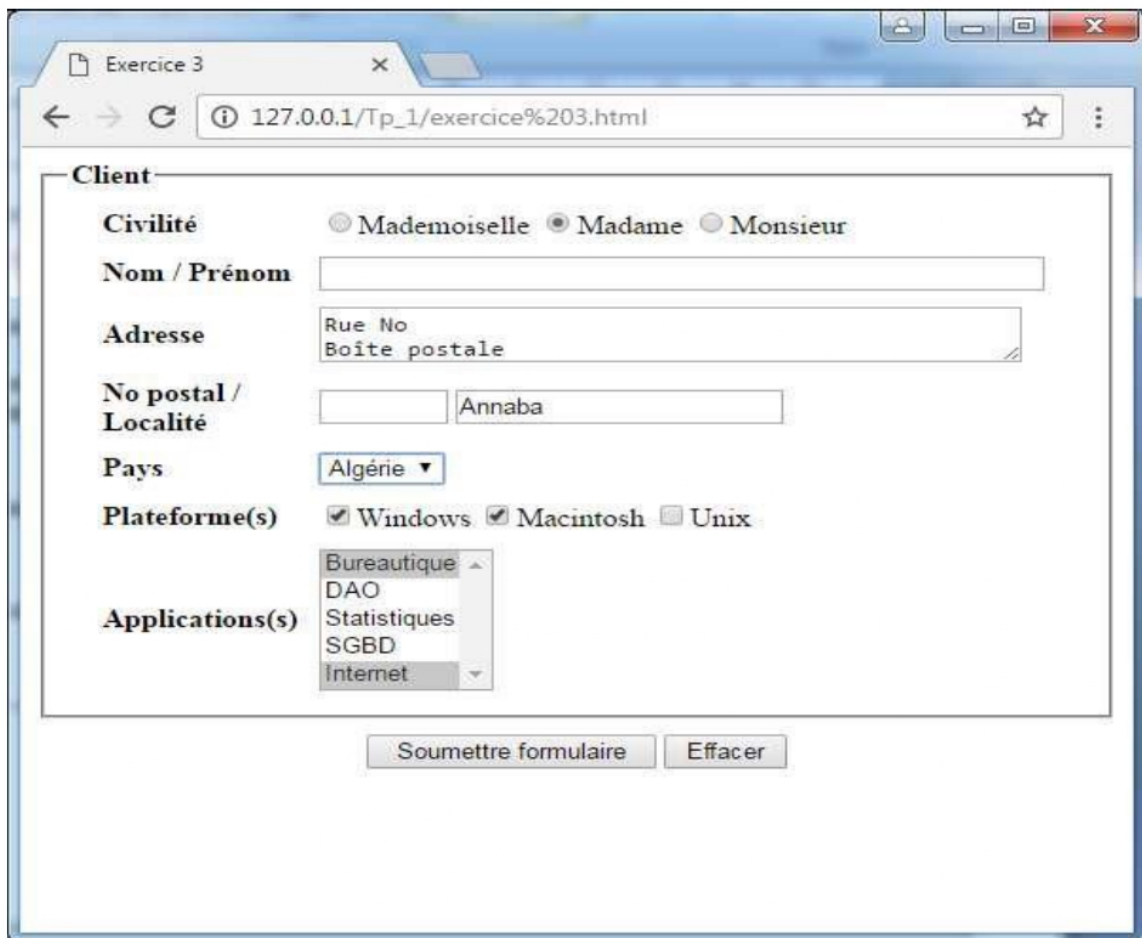
*Remarque : Pour chacune des pages, vous organiserez le contenu de la manière suivante en exploitant les balises h1 et h2 pour les titres et les sous-titres*





### 3.2 Exercice 2

Réaliser le formulaire ci-dessous :



The image shows a web browser window with the title 'Exercice 3' and the address bar containing '127.0.0.1/Tp\_1/exercice%203.html'. The main content is a form titled 'Client' with the following fields and controls:

- Civilité:** Radio buttons for 'Mademoiselle', 'Madame' (selected), and 'Monsieur'.
- Nom / Prénom:** A single-line text input field.
- Adresse:** A multi-line text area containing 'Rue No' and 'Boîte postale'.
- No postal / Localité:** Two text input fields, the second containing 'Annaba'.
- Pays:** A dropdown menu showing 'Algérie'.
- Plateforme(s):** Checkboxes for 'Windows' (checked), 'Macintosh' (checked), and 'Unix' (unchecked).
- Applications(s):** A dropdown menu with options: 'Bureautique', 'DAO', 'Statistiques', 'SGBD', and 'Internet'.

At the bottom of the form are two buttons: 'Soumettre formulaire' and 'Effacer'.

# Chapitre 3

## XML (eXtensible Markup Language)

## 1 XML, C'est quoi ?

Le langage XML (eXtended Markup Language) est un format général de documents orienté texte. Il s'est imposé comme un standard incontournable de l'informatique. Il est aussi bien utilisé pour le stockage de documents que pour la transmission de données entre applications. Sa simplicité, sa flexibilité et ses possibilités d'extension ont permis de l'adapter à de multiples domaines allant des données géographiques au dessin vectoriel en passant par les échanges commerciaux. De nombreuses technologies se sont développées autour de XML et enrichissent ainsi son environnement.

Le langage XML dérive de SGML (Standard Generalized Markup Language) et de HTML (HyperText Markup Language). Comme ces derniers, il s'agit d'un langage orienté texte et formé de balises qui permettent d'organiser les données de manière structurée. (Figure 20)

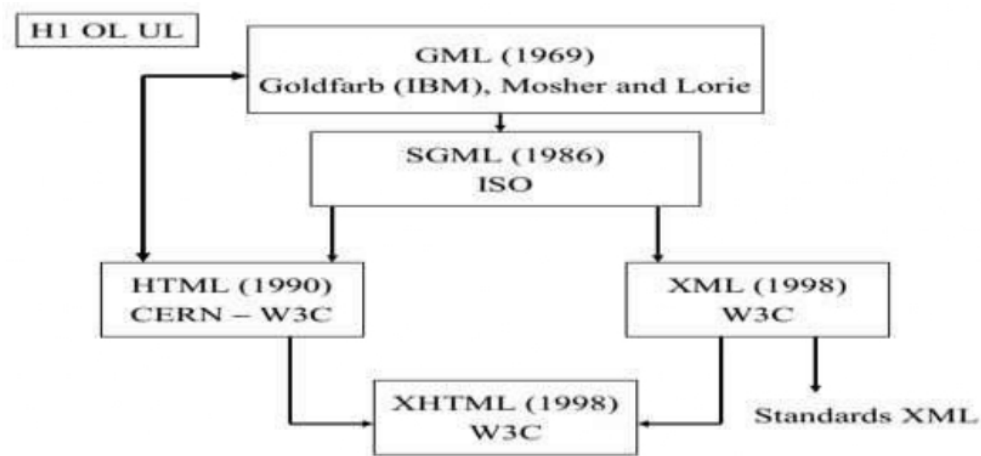


Figure 20: Origines de XML

### PS : XML n'est pas

1. Un langage de programmation.
2. Un protocole de transport de données.
3. Une base de données.
4. Le remplaçant de HTML.

## 2 XML et HTML

Paradoxalement, HTML est victime de son succès. Conçu pour un usage relativement simple (lien entre documents, présentation statique de pages web), il a du mal à s'adapter aux nouveaux besoins.

HTML présente en effet un certain nombre de faiblesses, c'est un langage :

- Non extensible (pas d'ajout dynamique de nouvelles fonctions),
- Non interactif,
- Orienté présentation,
- Non hiérarchique (pas de structuration des pages et des informations),
- Exploitable uniquement par les navigateurs,
- Mélangeant l'aspect présentation et le contenu,

- Dont les extensions sont propriétaires.

XML et HTML sont complémentaires, puisque XML traite des données, et que HTML les affiche. XML ne remplace donc pas HTML.

### 3 Positionnement de XML par rapport à SGML

SGML est un peu « l'ancêtre » de XML. Comme XML, il visait à séparer le contenu d'un document de sa structure et de sa présentation, mais il n'a pas connu un grand succès en raison de sa complexité et de son coût. Il est donc resté cantonné à certains secteurs d'activités spécifiques tels que l'aéronautique ou l'industrie automobile. XML aura sans doute dans un premier temps un usage limité à la gestion de documents puis ensuite un usage plus large (commerce électronique...). XML va vraisemblablement coexister avec SGML puis se substituer progressivement.

## 4 Principe et apport

### 4.1 Principes

- Documents lisibles par l'homme (Lecture et compréhension par l'homme, Rédaction dans un simple éditeur de texte).
- Conception rapide.
- Conception formelle et concise.
- Facilité de création.
- Peu de concision des balises.

### 4.2 Apports de XML

- Composants logiciels normalisés, réutilisables (Indépendance totale des fabricants de logiciels).
- Multiples vues des données.
- Traitement local des données.
- Recherches facilitées.
- Applications distribuées communicantes faciles à implémenter.

## 5 Syntaxe et structure

La syntaxe de XML est relativement simple. Elle est constituée de quelques règles pour l'écriture d'un entête et des balises pour structurer les données. Ces règles sont très similaires à celles du langage HTML utilisé pour les pages WEB mais elles sont, en même temps, plus générales et plus strictes. Elles sont plus générales car les noms des balises sont libres. Elles sont aussi plus strictes car elles imposent qu'à toute balise ouvrante corresponde une balise fermante. Il peut être stocké dans un fichier et/ou manipulé par des logiciels en utilisant un codage des caractères. Un document XML est généralement contenu dans un fichier texte dont l'extension est *.xml*.

XML, langage de balisage :

- Structure arborescente
- Éléments
- Attributs
- Couples nom/valeur associés aux éléments
  - Contenu et noms (éléments, attributs, valeurs) exemple :

On commence par donner un premier exemple de document XML comme il peut être écrit dans un fichier *bibliography.xml*. Ce document représente une bibliographie de livres sur XML. Il a été

tronqué ci-dessous pour réduire l'espace occupé. Ce document contient une liste de livres avec pour chaque livre, le titre, l'auteur, l'éditeur (publisher en anglais), l'année de parution, le numéro ISBN et éventuellement une URL. (Figure 21)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- Time-stamp: "bibliography.xml 3 Mar 2008 16:24:04" -->
<!DOCTYPE bibliography SYSTEM "bibliography.dtd" >
<bibliography>
  <book key="Michard01" lang="fr">
    <title>XML langage et applications</title>
    <author>Alain Michard</author>
    <year>2001</year>
    <publisher>Eyrolles</publisher>
    <isbn>2-212-09206-7</isbn>
    <url>http://www.editions-eyrolles/livres/michard/</url>
  </book>
  <book key="Zeldman03" lang="en">
    <title>Designing with web standards</title>
    <author>Jeffrey Zeldman</author>
    <year>2003</year>
    <publisher>New Riders</publisher>
    <isbn>0-7357-1201-8</isbn>
  </book>
</bibliography>
```

Figure 21: Exemple en XML

## 5.1 DTD (Document Type Definition)

- Contient la définition des : *Éléments, attributs de ces éléments, etc.*
- Fournit une structure type de document
- Permet de vérifier la validité des documents
- Constitue la grammaire historique de XML. exemple :

Cet exemple contient une déclaration de DTD qui permet de valider le document. Cette DTD déclare l'élément simple avec un contenu purement textuel.(Figure 22)

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<!DOCTYPE simple [
  <!ELEMENT simple (#PCDATA)>
]>
<simple>Un exemple simplissime</simple>
```

Figure 22: Exemple DTD

## 5.2 Document Bien Formé et Document Valide

### 5.2.1 Document bien formé :

Obéissant aux règles syntaxiques d'XML. Tout élément est inclus dans son père. Un unique élément racine

**5.2.2 Document valide :** Document bien formé, qui obéit à une structure type définie explicitement dans une DTD

- La validation fournit une interface garantie :
- Les producteurs valident pour assurer qu'ils livrent ce qu'ils promettent
- Les consommateurs valident pour vérifier ce qu'ils reçoivent
- Les auteurs d'application valident pour simplifier leur tâche
- Deux approches :
  - La *DTD* : définition simple et concise (pas de typage)
  - Les *XML Schémas* : définition plus complète, mais également plus complexe à mettre en œuvre.

### 5.3 Structure d'un document valide

- Un prologue facultatif
- Une définition d'une DTD facultative
- L'arbre d'éléments
- Ensemble de balises structurées sous forme arborescente
- Sections littérales
- Entités prédéfinies
- Commentaires et autres instructions de traitement facultatives

#### 5.3.1 Prologue

- Déclaration XML (**Figure 23**)
- La version du langage utilisée
- La déclaration d'encodage
- La présence ou non d'une DTD externe

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

Figure 23: Prologue

L'attribut *standalone* indique la présence ou non d'une *DTD externe*, il peut donc prendre deux valeurs :

- ' **no** ' si une DTD externe lui est associée. Dans ce cas, la DTD en question est déclarée grâce aux mots clés DOCTYPE et SYSTEM
- ' **yes** ' si le document se suffit à lui-même, et ne possède donc pas de DTD externe. Le document peut cependant être validé par une DTD interne.

La *déclaration d'encodage*, réalisée avec *encoding*, identifie le jeu de caractères utilisés ; comme par exemple **UTF-8**. Les analyseurs syntaxiques de XML sont tenus de supporter certains schémas **Unicode**, permettant le support de la plupart des langues.

#### PS

- Le Prologue est *facultatif*, mais l'attribut version est obligatoire
- Aucun autre attribut n'est autorisé

### 5.3.2 Déclaration de type de document

Le mot clé **DOCTYPE** : Spécifie le type de document et Permet l'appel d'une DTD *externe* (Figure 24)

```
<!DOCTYPE annuaire SYSTEM "exo2a.dtd" >
```

Figure 24: DTD externe

ou l'implémentation de celle-ci en *interne* (Figure 25)

```
<!DOCTYPE annuaire [  
  <!ELEMENT annuaire (#PCDATA) >  
>
```

Figure 25: DTD interne

On préfère en général une DTD externe à une interne. En effet en interne, on perd la réutilisabilité de la DTD, alors que cela représente, comme le verra, un avantage important.

### 5.3.3 Les éléments

- Balise d'ouverture (Figure 26)
- Contenu
- Balise de fermeture

```
<nomElem>contenu</nomElem>
```

Figure 26: Les éléments

Exception : éléments vides (Figure 27)

```
<nomElem/>
```

```
<nomElem></nomElem>
```

Figure 27: Eléments vides

#### • Règles d'utilisation :

- Tout élément fils est inclus dans son père
- Un unique élément racine
- Les éléments sont « case sensitive »

#### • Règles de nommage :

- Caractères alphanumériques et signes '-' '\_' '.'
- Caractères accentués éèèâ ...
- Premier caractère alphabétique ou signe '\_'

Pas d'espace

Ne débutant pas par la chaîne xml ou XML

L'élément racine est le seul et unique élément père qui contienne tous les autres.

#### Préconisation :

Outils de l'internet I

Dr YAKOUBI Mohamed Amine

Université B-M Annaba 25

- Éviter le ':' réservé aux noms de domaines
- Éviter les accents
- Éviter les '-' et '.'
- Démarrer le nom d'un élément avec une majuscule **Tests!!!**

Le document suivant est-il bien formé ? (Figure 28)

```
<groupe>
  <instrument>guitare
</instrument>
  <instrument>batterie
</groupe> </instrument>
<groupe>
  <instrument>orgue
</groupe></instrument>
```

Figure 28: Exemple

Ce document *n'est pas bien formé* car  
Il possède deux éléments racine  
L'élément `<instrument>` étant le *fil*s de `< groupe >`, il doit être *fermé avant lui*

### 5.3.4 Attributs

Les éléments peuvent avoir des attributs

- Chaque attribut est propre à un élément
- Les attributs décrivent certaines propriétés de l'élément
- Paire (Nom, valeur) Exemple : (Figure 29)

```
<article numero="1" prix="99"/>
```

Figure 29: Attributs

### 5.3.5 Mot clé CDATA

Déclaration d'une section littérale

Peut tout contenir, sauf `]]`, qui marque la fin Exemple: (Figure 30)

```
<element>
  <![CDATA[ <tag>contenu</tag> ]]>
</element>
```

Figure 30: CDATA

La réalisation de l'élément donnera : (Figure 31)

```
<tag>contenu</tag>
```

Figure 31: Exemple

### 5.3.6 Entité prédéfinies

Voir **Tableau 2**

*Tableau 2: Entités prédéfinies*

Caractère	Entité
<	&lt;
>	&gt;
&	&amp;
'	&apos;
"	&quot;

### 5.3.7 Commentaires

Voir **Figure 32**

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

*Figure 32: Commentaire*

### 5.3.8 Recommandations

*Règle 1 :*

Inclure des méta-données descriptives

*Règle 2 :*

Ne pas imposer le type de mise en valeur d'un élément (indépendance physique)

*Règle 3 :*

Bonne utilisation des attributs

Les balises de méta-données vous permettent de définir, par exemple, des éléments propres au document, comme la dernière date de modification, la catégorie du document, l'auteur, etc...

XML permet de dissocier forme et contenu. Il faut donc proscrire toute description visant à mettre en valeur un élément, par exemple par la création de tag "gras" ou "italique".

L'utilisation des attributs doit être un choix mûrement réfléchi. En effet, en fonction de la structure du document, et donc de la présence ou non d'attributs, le traitement des données sera différent. Il convient donc de réfléchir préalablement à la future utilisation du document.

#### ➤ **Elément Vs Attributs**

Voir **Tableau 3**

*Tableau 3: Elément vs Attributs*

Elément	Attributs
L'information représente une partie constituante de l'élément parent	L'information est inhérente à l'élément parent mais elle n'en fait pas partie

Représenter des données qui ne sont pas unique	Représenter des données uniques
Représenter du contenu	Représenter des relations entre des parties d'information

### 5.4 DTD

Passer d'un document bien formé à un document valide

Un document valide est une instance de la structure définie dans une DTD

- Factoriser les connaissances sur un document
- Approche minimaliste

Par rapport à l'effort requis avec XML Schema

- Réutilisabilité
- Permet d'être sûr du document reçu

#### 5.4.1 ELEMENT

Voir **Tableau 4**

*Tableau 4: ELEMENT*

Ordre	Symbole	Description
1	<	Délimiteur de début de balise
2	!	Présence d'un mot-clé
3	ELEMENT	Mot-clé de déclaration d'éléments
4	Nom	Nom de l'élément
5	Modèle	Type de contenu de l'élément
6	>	Délimiteur de fin de balise

- **Élément vide**

```
<!ELEMENT ANNUAIRE EMPTY >
```

```
<ANNUAIRE/>
```

- **Élément contenant du texte**

```
<!ELEMENT ANNUAIRE (#PCDATA) >
```

```
<ANNUAIRE>du texte</ANNUAIRE>
```

- **Avec contenu non défini**

```
<!ELEMENT ANNUAIRE (ANY) >
```

**5.4.2 Contenu des éléments**

- **Éléments fils**

Peuvent avoir un ordre imposé et former une séquence.

- Séquence spécifiée par la liste des noms des éléments.
- Séparateurs

*Tableau 5: Séparateur*

Séparateur	Description
,	Enchaînement fixe
	Choix parmi plusieurs possibilités

**PS :** à partir du **Tableau 5** nous pouvons dire que

- ✓ La virgule est un caractère de séparation signifiant littéralement "ET". Dans ce cas, l'ordre d'apparition des éléments est important.
- ✓ Le signe "|" est un caractère de séparation signifiant littéralement "OU".

- **Élément avec un sous-élément**

```
<!ELEMENT EMPLOYE (NOM) >
<!ELEMENT NOM (#PCDATA) >

<EMPLOYE>
  <NOM>GARNIER</NOM>
</EMPLOYE>
```

- **Élément avec plusieurs sous-éléments**

```
<!ELEMENT EMPLOYE (NOM, PRENOM) >
<!ELEMENT NOM (#PCDATA) >
<!ELEMENT PRENOM (#PCDATA) >

<EMPLOYE>
  <NOM>GARNIER</NOM>
  <PRENOM>EMMA</PRENOM>
</EMPLOYE>
```

**5.4.3 Occurrence des éléments**

- **Élément fils**

*Tableau 6: Indicateurs*

Indicateur	Description
	Obligatoire
?	Optionnel non répétable
*	Optionnel et répétable
+	Obligatoire et répétable

*Exemple :*

```
<!ELEMENT CHAPITRE (TITRE, SECTION+) >  
<!ELEMENT LIVRE (TITRE, (CHAPITRE,TITRE)+, ANNEXE*) >
```

*Question* : le document XML ci-dessous est-il validé par la DTD suivante ?

```
<!ELEMENT groupe (instrument+,style)*>
```

```
<groupe>  
  <style>Rock</style>  
  <instrument>guitare</instrument>  
  <instrument>batterie</instrument>  
</groupe>
```

*Réponse* : NON : style doit être APRES instrument, et non avant

Un document **valide** pourrait être :

```
<!ELEMENT groupe (instrument+, style)>
```

```
<?xml version='1.0' standalone='no' ?>  
<!DOCTYPE groupe SYSTEM 'el.dtd' >  
<groupe>  
  <instrument>guitare</instrument>  
  <instrument>batterie</instrument>  
  <style>Rock</style>  
</groupe>
```

### *Exercice*

Écrire les DTDs permettant de valider un document XML avec la contrainte suivante :

- Un groupe doit posséder au moins deux instruments.
- Un concert doit posséder soit aucun groupe soit exactement deux groupes.

*Correction* :

- Un groupe doit posséder au moins deux instruments

```
<!ELEMENT groupe (instrument, instrument+) >
```

- Un concert doit posséder soit aucun groupe soit exactement deux groupes

```
<!ELEMENT concert (groupe, groupe)? >
```

#### 5.4.4 Attributs

Le mot-clé ATTLIST :

*Intérêts*

- Déclarer des attributs obligatoires ou non.
- Leur donner une valeur par défaut

*Syntaxe :*

```
<!ATTLIST nom-élément
          nom-attribut type-attribut val-défaut>
```

*Exemple :*

```
<!ELEMENT Employe EMPTY>
<!ATTLIST Employe nom CDATA "non spécifié">
```

#### 5.4.5 Types des attributs

Voir **Tableau 7**

*Tableau 7: Attributs*

Type	Description
CDATA,	Chaîne de caractères littérale
ID ou IDREF	Renvoi à l'intérieur des documents
ENTITY	Nom d'une entité externe XML
NMTOKEN	Nom symbolique quelconque formé de caractères alphanumériques
NOTATION	Attribut de notation
Liste de choix	Choix entre plusieurs Tokens

- **Contrainte de validité : ID** Les valeurs ID doivent identifier uniquement les éléments qu'elles désignent (Elles ne peuvent apparaître plus d'une fois dans un document XML).
- **Contrainte de validité : Un seul ID par élément** Aucun élément ne peut posséder plus d'un attribut ID.
- **Contrainte de validité : Valeur implicite de l'attribut ID** Un attribut ID doit avoir comme valeur implicite déclarée soit #IMPLIED soit #REQUIRED.
- **Contrainte de validité : IDREF** Les valeurs IDREF doivent correspondre à la valeur d'un attribut ID. **Les attributs énumérés** peuvent prendre une valeur parmi une liste de valeurs fournies dans la déclaration.

- **Déclaration par défaut**

Voir **Tableau 8**

*Tableau 8: Déclaration par défaut*

Type	Description
------	-------------

' <i>valeur</i> '	Valeur par défaut
#REQUIRED	Attribut obligatoire
#IMPLIED	Attribut facultatif
#FIXED ' <i>valeur</i> '	Attribut à valeur fixe

*Exemple*

```
<!ELEMENT Employe (#PCDATA) >
<!ATTLIST Employe nom CDATA #REQUIRED
pre nom CDATA #IMPLIED
agence CDATA #FIXED "RENNES"
sexe (homme|femme) "femme">
```

- Dans les documents valides, les attributs **xml:lang** et **xml:space** doivent être déclarés s'ils sont utilisés

*Exemple pour xml:lang*

```
<!ATTLIST poésie xml:lang NMTOKEN 'fr'>
```

*Exemple pour xml:space*

```
<!ATTLIST poème
xml:space (default|preserve) 'preserve'>
```

**5.4.6 Entités**

Permettent de déclarer des blocs d'information

Elles sont utilisées dans

la DTD => *entités paramètres*

le document XML => *entités générales*

Elles peuvent être

Internes => *le contenu suit sa déclaration*

Externes => *dans les autres cas*

- **Entité paramètre** : déclaration dans la DTD et utilisation dans la DTD

*Exemple*

```
<!ENTITY % Copyright "Copyright Softeam">
<!ENTITY % Soft "Objecteering 431 - %Copyright;">
```

- **Entité générale** : déclaration dans la DTD et utilisation dans l'instance.

*Exemple*

```
<!ATTLIST image href ENTITY #REQUIRED>
<!ENTITY photol SYSTEM "toto.gif">
```

```
<image href="photol"/>
```

Les entités générales sont utilisées dans l'instance des documents pour stocker des données non XML et pour modulariser un document.

Les entités paramètres sont utilisées dans la DTD pour paramétrer une DTD ou des déclarations de balisage

- **Entités internes** : Précédées par le caractère ' & ' et Terminées par le caractère ' ; '

*Exemple*

```
<!DOCTYPE test [
  ...
  <!ENTITY ref "Softeam" >
]>
<test>
  <p>&ref;</p>
</test>
```

Les entités internes permettent généralement de ne pas réécrire une chaîne de caractères plusieurs fois. Lors de l'analyse du document, la référence est remplacée par sa valeur.

Ces entités internes sont déclarées grâce au mot-clé ENTITY, dans le corps de la déclaration de type.

Les entités internes ainsi déclarées sont ensuite utilisables grâce au caractère ' & ' dans le corps du document. Il suffit simplement de faire précéder le nom de l'entité utilisée par le caractère ' & ', et de la terminer par un point-virgule.

- **Entités externes adressées par URL** Mot clé SYSTEM et les entités externes adressées par URL

*Exemple*

```
<!ENTITY nom PUBLIC FPI URL>
```

```

<!DOCTYPE bouquin [
  <!ENTITY chapitre-1 SYSTEM "./chap1.xml">
  <!ENTITY chapitre-2 SYSTEM "./chap2.xml">
  ...
]>

<bouquin>
  &chapitre-1;
  &chapitre-2;
</bouquin>

```

Les entités externes sont en fait des documents XML bien formés, qui peuvent être utilisés à l'intérieur d'un autre document XML. Lors de la réalisation du document, ces entités sont remplacées par leur contenu.

- **Entités externes adressées par FPI**

Mot-clé **PUBLIC**

```
<!ENTITY nom PUBLIC FPI URL>
```

*Nom* : nom donné à l'entité.

*FPI* : Formal Public Identifier.

*URL* : Unified Ressource Location.

L'*URL* est utilisée si l'application n'est pas capable de résoudre l'identificateur publique.

- **Un identificateur FPI** est composé de quatre parties séparées par des `"/"` :

```

<!ENTITY rec-XML PUBLIC
  "-//W3C//DOCUMENT Recommendation open-
  The recommandation for XML 1.0//EN"
  " http://www.W3.org/TR/1999/REC-xml-
  19992312.xml">

```

- Type d'identificateur
  - Identificateur enregistré (+)
  - Non enregistré (-)

Les identificateurs de ce type sont très utiles en SGML car ils identifient de façon unique des ressources normalisées (tables de caractères ou de symboles). Ils sont moins utiles avec XML du fait que le jeu de caractères standard de XML est lui-même une norme ISO.

- Propriétaire de l'identificateur public

Ne s'applique pas aux identificateurs ISO

Texte public, composé d'un mot-clé, et d'une description textuelle libre

DOCUMENT DTD ENTITIES SHORTREF SUBDOC NOTATION Etc.
---

- **Notion d'entités locales (Entité paramètre)**

Précédées par le caractère ' % '

Référencées uniquement à l'intérieur d'une DTD

*Exemple*

```
<!ENTITY % nl "numérotée">
<!ENTITY % ol "ordonnée">

<!ELEMENT liste EMPTY>
<!ATTLIST liste type (%nl; | %ol;) "%nl;">
```

**PS : NDATA** qualifie une entité non XML. Une entité non xml ne peut être utilisée dans une DTD. C'est pourquoi l'association NDATA avec une entité paramètre est impossible.

#### 5.4.7 Règles d'écriture

➤ *Règle 1*

Modularité, pour favoriser la réutilisabilité.

➤ *Règle 2*

Regrouper les déclarations d'entités paramètres.

➤ *Règle 3*

Utilisation des entités paramètres.

Pour la déclaration d'attributs pour les modèles de contenu.

➤ *Règle 4*

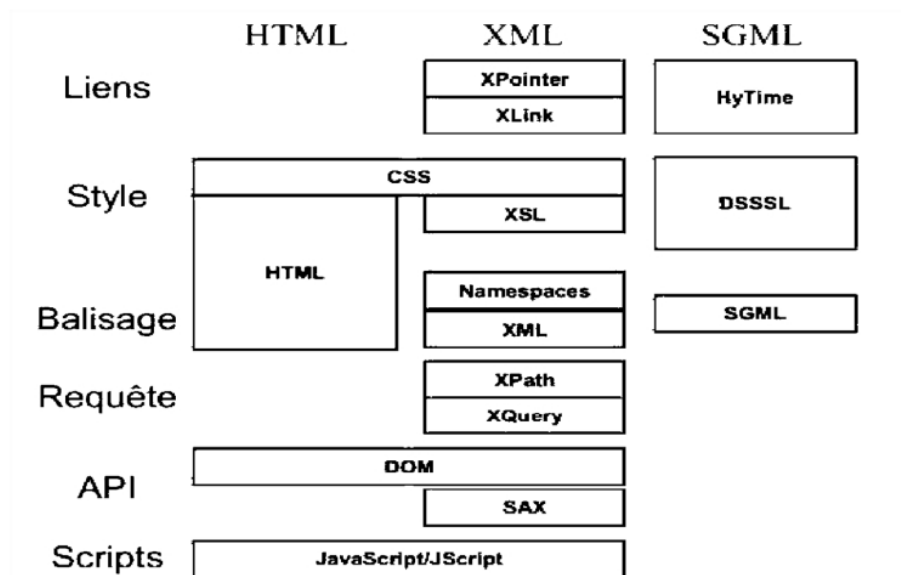
Simplicité des modèles de contenu.

```
<!ELEMENT alinea
(paragraphe|liste|tableau|bmp|cgm|eps|mpeg|note)*>
```

```
<!ELEMENT alinea
(paragraphe|liste|tableau|note|illustration)*>
<!ELEMENT illustration
(image|graphique-cgm|graphique-eps|video)>
```

- Règle 5  
Bonne utilisation des commentaires.

### 5.8 Standards liés à XML



- **Langage DSSSL** : Document Style, Semantic and Specification Language. (langage de style, de sémantique et de spécifications de document) (ISO 10179) Une norme associée au langage SGML qui précise les règles d'un langage ouvert. Afin de régir l'apparence et le style des composantes logiques (p. ex., en-têtes de chapitres), définies par le langage SGML.

Les standards sont découpés en modules pouvant éventuellement être regroupés lors de versions ultérieures.

- **XML 1.0 (rec)** : module de base.
- **Namespaces (rec)** : module de différenciation des éléments et attributs synonymes en leur rajoutant un préfixe.
- **XPath (rec et req)** : langage permettant d'identifier des sous-ensembles dans un document.
- **DOM (rec, rec & WD)** : Document Object Model -- interface OO indépendante des langages.
- **Infoset (WD)** : description des informations disponibles dans un document XML.
- **XSL(CR) et XSLT (rec, et req)** : langages de transformation et présentation d'un document.

- **CSS2 (rec)** : définition de style pour documents XML.
- **XML Schema (Rec)** : définition de schémas XML.
- **Xinclude (LC), Xpointer (LC), XMLBase (PR) et Xlink (PR)** : famille de recommandations gérant les liens et inclusions entre documents.
- **XQuery (WDs)** : langage d'interrogation.
- **XPATH** : est un langage d'expression s'appliquant à un document XML. Il est destiné à la recherche de nœuds dans un arbre XML. Il Suit une syntaxe simple et non ambiguë. Il Manipule des nœuds et des ensembles de nœuds
- **XSL eXtensible Stylesheet Language**. XSL est un langage de feuilles de styles
- **XSLT** : la partie transformation
- **XSL-FO** : un jeu d'instructions de formatage en XML destinées à la présentation
- **SAX** : Simple Api for XML. SAX est une interface de programmation "simple" pour parser XML. Elle est basée sur un modèle événementiel  
`startDocument(), endDocument() startElement(), endElement() characters(), etc.`  
Elle est surtout utilisée par les parsers en Java
- **DOM** : Un modèle de document. Une interface de programmation indépendante du langage et des plates-formes pour manipuler :
  - La structure des documents*
  - Le contenu des documents*Aussi bien pour des documents XML que HTML ○ Une recommandation du W3C
- **XQuery** : est une spécification destinée à la recherche d'information. Proposée par IBM et les auteurs de XML/QL. Soutenue par Microsoft

Forme de requête simple

```
FOR $<var> in <forest>[, $<var> in <forest>] //itération  
  
LET $<var> := <subtree> // assignation  
  
WHERE <condition> // élagage  
  
RETURN <result> // construction
```

- **XML Schema** : Un langage de schéma. Au même titre que les DTD constituent le langage de schéma de SGML

Les XML schémas permettent

- De décrire la structure des informations
- D'avoir un mécanisme de validation de texte
- 100 % XML
- Une recommandation du W3C

## 6 Parseurs XML

- ✓ Indispensable pour les applications utilisant XML
- ✓ Lit, analyse les DTDs
  - Déclaration d'éléments, attributs, entités, notation
  - Détection des ambiguïtés

- ✓ Lit, analyse les instances
  - Structuration arborescente des éléments
  - Références aux entités
- ✓ Microsoft XML Parser (Microsoft)
  - Plate-forme : Windows
- ✓ Xerces (Projet Apache)
  - Plate-forme : Java / C++ / Perl
- ✓ XP (James Clark)
  - Plate-forme : Java (JDK 1.1)
- ✓ PHP XML Parser (Manuel Lemos)
  - Plate-forme : PHP

## 7 Editeurs XML

- ✓ Visual DTD (Alphaworks, IBM)
  - Plate-forme : Java
- ✓ Near & Far Designer (Microstar)
  - Plate-forme : Windows 95/98/NT
- ✓ ttdt (Mulberry Technology)
  - Plate-forme : Emacs
- ✓ XML Spy (Altova)
  - Plate-forme: Windows 98/NT/2000/XP

## 8 Exercices

### 8.1 Exercice 1

1. Ecrire un document XML qui permet de représenter un algorithme. Ce dernier peut contenir une partie déclaration et une partie corps. La première est constituée d'une ou plusieurs variables et la deuxième est composée d'une ou plusieurs instructions.

2. Vérifier que votre document est bien formé.

### 8.2 Exercice 2

On souhaite ajouter des attributs au document de l'exercice précédent et ce comme suit :

1. Auteur, dateModif et version pour la balise algorithme.
2. Type pour les variables.
3. Et Type pour les instructions.

Ecrire un document XML qui permet d'avoir ces informations.

### 8.3 Exercice 3

1. Créez la DTD carnet.dtd suivante :

```
<!ELEMENT produit (client+)>
```

```
<!ELEMENT client EMPTY>
```

```
<!ATTLIST client
  nom CDATA #REQUIRED prenom
  CDATA #IMPLIED telephone
  CDATA #REQUIRED>
```

2. Créez un document XML qui soit valide par rapport à cette DTD

#### 8.4 Exercice 4

Générer par programmation un fichier XML contenant les éléments suivants :

```
<configuration>
  <database>
    <username>dbuser</username>
    <password>secret</password>
  </database>
  <database>
    <username>root</username>
    <password>root</password>
  </database>
</configuration>
```

#### 8.5 Exercice 5

Nous voulons sauvegarder des informations sur des rectangles qu'on dessine sur l'écran. Chaque rectangle a les informations suivantes : x (position sur l'axe horizontale), y (position sur l'axe vertical), hauteur, largeur et une couleur (rouge, vert, etc).

- 1- Ecrire un document XML qui permet de stocker ces informations.

# Chapitre 4

## Courrier électronique

## 1 Introduction

Le courrier électronique est aujourd'hui l'une des applications les plus populaires du réseau. Utilisé pour des applications très variées - personnelles, professionnelles, associatives, politiques, etc. -, celui-ci tend à prendre une place de plus en plus prépondérante par rapport aux moyens de communication traditionnels.

## 2 Courrier électronique ?

Mail, e-mail, courrier électronique, courriel : c'est la même chose ! Ces termes désignent un message échangé entre utilisateurs via Internet.

Le courrier électronique est un service de transmission de messages écrits et de documents envoyés électroniquement via le réseau Internet dans la boîte aux lettres électronique d'un destinataire choisi par l'émetteur.

Comme une lettre postale traditionnelle, ce courrier électronique comporte :

- Une adresse (électronique) du ou des destinataires principaux (et éventuellement d'autres destinataires en copie) ;
- Un contenu c'est à dire le message à proprement parler. C'est en général un texte auquel vous pouvez associer un fichier, des photos, images, musique,...etc.
- Quelques informations destinées à l'acheminement du message, équivalentes des tampons postaux (date et heure, etc.)

## 3 Fonctionnement d'un email ?

Pour émettre et recevoir des messages par courrier électronique, il faut disposer d'une adresse électronique et d'un client de messagerie ou d'un webmail permettant l'accès aux messages via un navigateur Web.

Sur Internet, il y a des serveurs (des gros ordinateurs qui fonctionnent 24h/24h) dédiés au courrier électronique. Chaque serveur fournit des adresses à des utilisateurs.

Les serveurs ont deux rôles :

- Ils se chargent d'expédier à une adresse se trouvant sur n'importe quel serveur de courrier dans le monde le courrier électronique envoyé par leurs utilisateurs ;
- ils stockent le courrier provenant de n'importe quel serveur dans le monde et destiné à leurs propres utilisateurs.

Les serveurs de courrier fonctionnent un peu comme les bureaux de poste dans le monde, sauf que deux serveurs de courrier s'envoient du courrier électronique en quelques secondes ou quelques dizaines de secondes.

## 4 Création d'un compte électronique

Pour pouvoir envoyer et recevoir des messages via Internet, vous devez avoir une adresse et une boîte aux lettres (comme pour le courrier normal).

Pour cela, vous devez créer (ouvrir) un compte de courrier électronique auprès d'un organisme qui dispose d'un serveur de courrier.

Différentes possibilités existent :

- Il existe des fournisseurs de courrier sur Internet qui proposent des adresses gratuites et payantes. Les adresses gratuites offrent un peu moins de possibilités que les adresses payantes mais suffisent largement pour un usage courant (Exemple : Outlook.be et Gmail sont des fournisseurs mondiaux de courrier électronique ; Gmail étant le fournisseur lié au fameux moteur de recherche Google).
- Les fournisseurs d'accès à Internet offrent aussi une adresse électronique à leurs clients (exemple : sandrine.ledoux@skynet.be). Il faut cependant être vigilant avec ces adresses car elles sont liées à votre fournisseur d'accès à Internet. En résiliant votre abonnement chez votre fournisseur, vous perdez donc votre adresse !
- Différents organismes (sociétés, administrations) donnent parfois une adresse de courrier électronique à leurs employés pour un usage professionnel (exemple : pierre.dupont@entreprise.com), ou des écoles à leurs étudiants, etc.

### 5 Adresse électronique

Les adresses électronique dans le service de messagerie (émetteurs ou destinataires) sont des couples séparés par le caractère « @ » (arobase) :

**utilisateur@serveur.domaine**

La partie de droite décrit le nom de serveur de messagerie concerné et la partie de gauche désigne l'utilisateur appartenant à ce serveur. A chaque domaine correspond un ou plusieurs serveurs de messagerie.

### 6 Messagerie électronique

La messagerie électronique s'appuie principalement sur des serveurs de messagerie, des protocoles de transport ainsi que sur des protocoles de contenu.

### 7 Serveurs de messagerie

La messagerie électronique n'est pas un service « point à point », ce qui signifie que les machines émettrices et réceptrices des messages n'ont pas besoin d'être reliées ensemble directement pour pouvoir communiquer.

Les messages sont transmis d'une machine à l'autre à travers le réseau Internet jusqu'à leur destination finale. Ces machines qui sont chargées d'acheminer et de réceptionner le courrier électronique sont appelées serveurs de messagerie.

Chaque e-mail est stocké sur un serveur avant d'être lu. Même si chaque boîte aux lettres est protégée par un identifiant et un mot de passe, la messagerie électronique apparaît, à cet égard, comme un service moins sécurisé et moins confidentiel que le courrier traditionnel.

## 8 Protocoles de communication

Un courrier électronique circule par définition sur Internet. Il passe d'ordinateur en ordinateur (ou de serveur en serveur), de l'ordinateur de l'expéditeur à celui du destinataire. Pour ce, il emprunte différentes voies et utilise plusieurs protocoles

### 8.1 Le protocole SMTP

Le protocole SMTP (Simple Mail Transfer Protocol) est le protocole standard permettant de transférer le courrier entre deux serveurs de messagerie - celui de l'expéditeur et celui du destinataire.

Un message met en général quelques secondes seulement pour aller d'un point à un autre sur l'Internet. Le message peut transiter par différents relais ou par un seul serveur si le destinataire utilise le même serveur de messagerie que l'expéditeur

### 8.2 Le protocole POP

Le protocole POP (Post Office Protocol) est encore aujourd'hui encore couramment utilisé.

Le protocole POP, dont la dernière version est la version 3 (on parle ainsi de POP3), a été conçu pour vous permettre de récupérer votre courrier sur une machine distante quand vous n'êtes pas connecté en permanence à Internet.

Le protocole POP gère l'authentification, c'est-à-dire la vérification de vos identifiant et mot de passe. Il bloque, également, votre boîte aux lettres pendant que vous y accédez, ne permettant pas à une autre connexion d'accéder en même temps à votre courrier.

### 8.3 Le protocole IMAP

L'IMAP (Interactive Mail Access Protocol) est un protocole qui vous permet, depuis un programme installé sur votre ordinateur ou votre smartphone, d'accéder aux messages de votre boîte aux lettres électronique.

IMAP est ainsi le protocole idéal pour ceux qui ont l'habitude d'utiliser leur messagerie de différentes manières, depuis leur ordinateur de bureau, leur ordinateur portable, leur tablette tactile ou leur smartphone. Les clients de messagerie peuvent lire, supprimer ou déplacer le courrier sur le serveur.

## 9 Conclusion

Le courrier électronique, comme toutes les applications d'Internet, repose sur des protocoles de communication

Les trois principaux protocoles utilisés par un serveur de messagerie sont le SMTP (Simple Mail Transfer Protocol), le POP (Post Office Protocol) et l'IMAP (Internet Message Access Protocol).

## 10 Exercices

### 10.1 Exercice 01

1. Quelle est la différence entre l'Internet et le Web ?
2. Donnez une petite définition pour un courrier électronique ?

3. C'est quoi un serveur de messagerie électronique ?
4. Citez les protocoles de contenu de la messagerie électronique et quelle est la différence entre eux ?

### **10.2 Exercice 02**

Ecrire un paragraphe de quelques lignes en expliquant les différentes étapes passées durant la transmission des courriers électroniques entre deux terminaux

## 6. Références

- 1- HTML et JavaScript pour les nuls, Emily A. Vander Veer edition First 2014.
- 2- XML par la pratique, Thierry BOULANGER, Edition ENI, 3-ème Edition 2015.
- 3- Bien débiter avec HTML, Rémy Lentzner, édition Remylent, 2020
- 4- Pour les Nuls : HTML & CSS3 Pour les Nuls, Ed Tittel, Jeff Noble, édition First Interactive, 2021.
- 5- L'essentiel de XML Cours XML Olivier Carton, Support du cours XML en M2 Pro à l'Université Paris Diderot, 2015.
- 6- Apprentissage XML: Le guide ultime du débutant pour apprendre la programmation XML étape par étape, Richard Baker, édition Amazon Digital Services LLC - KDP Print US, 2021.
- 7- XML: DTD, XML-Schema, XPath, XQuery, XSL-FO, SAX, DOM, Margit Becher, édition Springer Vieweg, 2021.
- 8- XSL-FO Understand and Use: XML Processing for PDF and Print, Manfred Krüger, édition MID/Information Logistics, 2021.
- 9- Internet. Une infographie, Tristan Mendes france, Quintin Leeds, édition CNRS, 2021.
- 10- Le courrier électronique avec Mail, Agnosys,, Franck Sartori, Thierry Deltit, édition Mon Mac & Moi, 2008.
- 11- Courrier électronique et comportements : usages et mésusages, Fernando Lagraña, édition London : Íste editions, cop. 2016.