

Université Abou-Bekr Belkaid –Tlemcen-
Faculté des Sciences
Département d'Informatique

Le langage XML **(eXtensible Markup Langage)**

Année 2018 - 2019

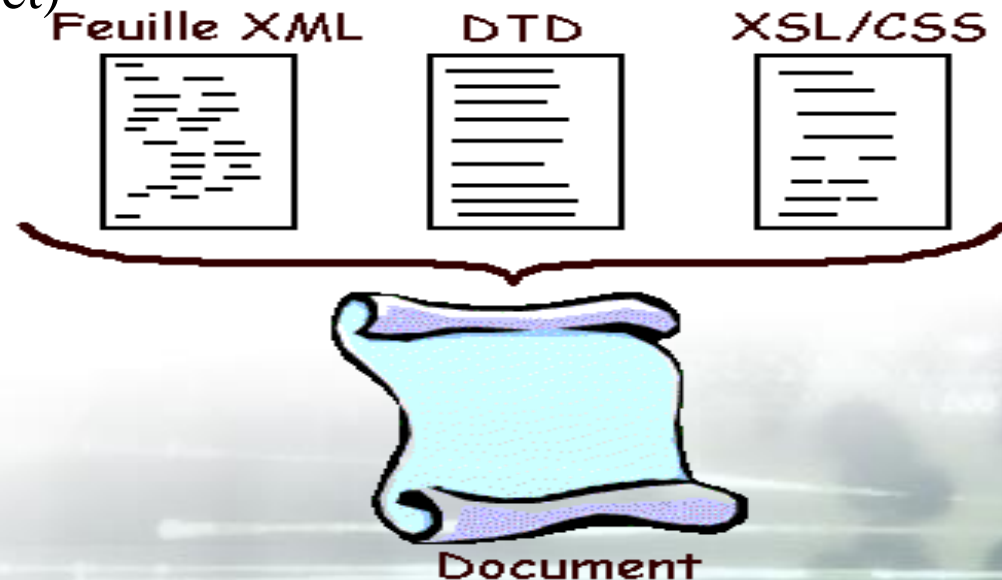
XML : eXtensible Markup Language

- Un langage de balisage extensible standardisé par le World Wide Web Consortium (W3C). XML est principalement connu comme la future norme d'échange de documents sur le WEB. Il peut être considéré comme une généralisation de HTML où, au lieu d'avoir un jeu de balises prédéfinies ayant une signification prédéfinie, l'auteur peut "inventer" ses propres balises. En réalité, sous ce sigle se cache un outil de description bien plus puissant reposant sur un langage générique (hérité de SGML).
- Le but de XML est de faciliter le traitement automatisé de documents et de données. L'idée est de pouvoir structurer les informations de telle manière qu'elles puissent être à la fois lues par des personnes sur le web et traitées par des applications qui exploiteront de manière automatisée les informations en question.

XML : buts

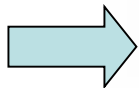
- XML= un nouveau langage d'échange basé sur le balisage
- XML= plus simple que SGML
- XML= plus complexe et performant que HTML
- XML = développé par XML Working Group dirigé par le W3C (depuis 1996)
- XML 1.0 = recommandation officielle du W3C depuis le 10 février 1998

- **XML** permet de séparer le contenu de la présentation :
Cette séparation fait d'un document XML un objet portable et réutilisable par un nombre illimité d'applications;
- Description de la structure d'un document XML via:
 - ❖ **DTD (Document Type Definition)**
 - ❖ **XML Schéma**
- La mise en page un document XML via:
 - ❖ **XSL (eXtensible Style Sheet Language)**
 - ❖ **CSS (Cascading Style Sheet)**



Les avantages de XML :

- **Lisibilité** : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML
- **Autodescriptif** : définir des balises de notre choix.
- Une **structure arborescente** : permettant de modéliser la majorité des problèmes informatiques
- **Universalité et portabilité** : les différents jeux de caractères sont pris en compte; format texte; il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme HTTP
- **Intégrabilité** : un document XML est utilisable par toute application pourvue d'un « *parser* » (c'est-à-dire un logiciel permettant d'analyser et d'interpréter un code XML)
- **Modularité, extensibilité** : un document XML doit pouvoir être utilisable dans tous les domaines d'applications; XML est un métalangage dont les bases peuvent être utilisées pour créer d'autres langages (XLink, ...)



XML est particulièrement adapté à l'échange de données et de documents

Les technologies liées à XML

- **CSS**, permet de définir une feuille de style pour XML (et HTML) ;
- **XSL**, langage évolué pour la définition de feuilles de style propre à XML ;
- **Xlink** pour ajouter des liens hypertextes à un fichier XML ;
- **XPointer** pour pointer sur des éléments de données d'un document XML ;
- **DOM *Document Object Model*** pour manipuler des fichiers XML (et HTML) à partir d'un langage de programmation;
- **Xquery** pour extraire des données des documents XML;
- **XForm** pour les formulaires.

Applications XML :

- XML est un métalangage permettant l'élaboration de nouveaux langages (balisages spécialisés). En fonction du domaine d'application dans lequel on souhaite publier, on définit des balises qui lui sont propres.
- Quelques applications dérivées de XML pour répondre aux besoins de plusieurs domaines:
 - ✓ **OFX** : **O**pen **F**inancial **eX**change pour les échanges d'informations dans le monde financier
 - ✓ **MathML** : **M**athematical **M**arkup **L**anguage permet de représenter des formules mathématiques
 - ✓ **CML** : **C**hemical **M**arkup **L**anguage permet de décrire des composés chimiques
 - ✓ **SMIL** : **S**ynchronized **M**ultimedia **I**ntegration **L**anguage permet de créer des présentations multimédia en synchronisant diverses sources : audio, vidéo, texte...
 - ✓ **SVG** : **S**calable **V**ector **G**raphics : futur standard permettant la création de graphiques vectoriels
 - ✓ **WML** : **W**ireless **M**arkup **L**anguage : internet mobile.
 - ✓ **AML** : **A**stronomical **M**arkup **L**anguage: langage décrivant les différents types de données utilisées en astronomie ;

Outils XML:

- XMLSpy est un très bon éditeur XML édité par Altova Inc.,
- Cooktop est un éditeur XML, XSLT, XPath et DTD puissant et totalement gratuit,
- XMetal est un logiciel puissant et flexible pour la création de documents XML,
- XMLWriter est un puissant éditeur XML,
- XMLMind représente un très bon éditeur XML,
- JXMLEditor est un éditeur XML écrit en Java,

Structure d'un document XML

- La norme XML en tant que telle doit être vue comme un outil permettant de définir un langage (on dit alors qu'il s'agit d'un métalangage),
- Une balise est une chaîne de caractère du type: <balise>

<annuaire>

<personne class = "etudiant">

<nom>Desjardins</nom>

<prenom>Jean-Philippe</prenom>

<telephone>(819) 234 2343</telephone>

<email>webmaster@usherbrooke.ca</email>

<!-- insertion de commentaires XML -->

</personne>

<personne>

...

</personne>

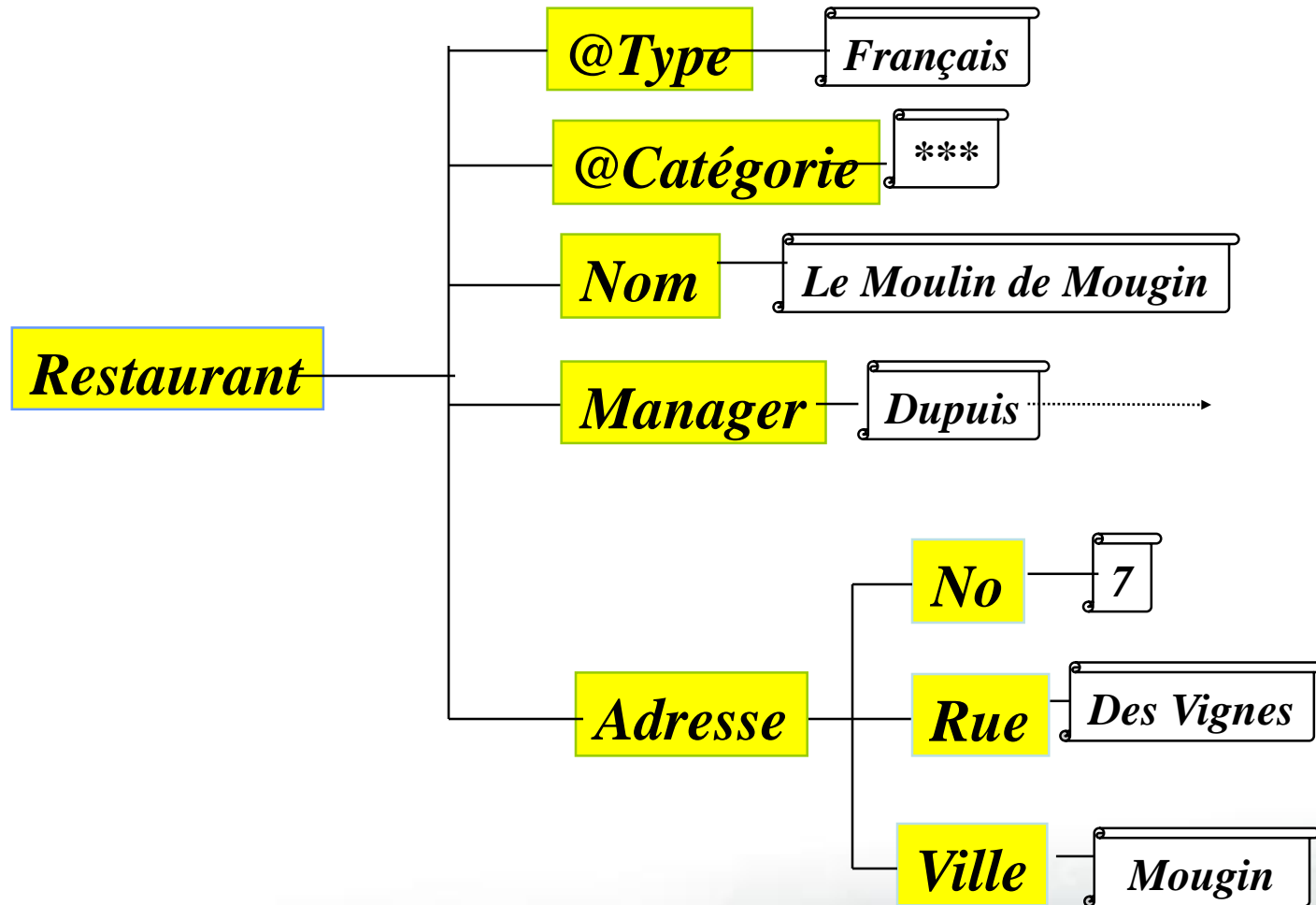
</annuaire>

Premier Exemple

```
<html>
  <head>
    <title>
      Livre informatique
    </title>
  </head>
  <body>
    "Java et Internet" par Gilles
    Roussel et Etienne
    Edition Duris Vuibert 600
    pages
  </body>
</html>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<livre>
  <titre>Java et Internet</titre>
  <auteur>
    <prenom>Etienne</prenom>
    <nom>Duris</nom>
  </auteur>
  <auteur>
    <prenom>Gilles</prenom>
    <nom>Roussel</nom>
  </auteur>
  <editeur>Vuibert</editeur>
  <taille>600</taille>
</livre>
```

Modèle arborescent



Présentation textuelle

<Restaurant Type="Français" Categorie="***">

<Nom>Le Moulin de Mougin</Nom>

<Manager ... >Dupuis</Manager>

<Adresse>

<No> 7</No>

<Rue>Des Vignes</Rue>

<Ville>Mougin</Ville>

</Adresse>

</Restaurant>

Autre Exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- La ligne ci-dessus est le prologue -->
<!-- Élément racine -->
<biblio>
  <!-- Premier enfant -->
  <livre>
    <!-- Élément enfant titre -->
    <titre>Les Misérables</titre>
    <auteur>Victor Hugo</auteur>
    <nb_tomes>3</nb_tomes>
  </livre>
  <livre>
    <titre>L'Assomoir</titre>
    <auteur>Émile Zola</auteur>
  </livre>
  <livre lang="en">
    <titre>David Copperfield</titre>
    <auteur>Charles Dickens</auteur>
    <nb_tomes>3</nb_tomes>
  </livre>
</biblio>
```

Structure d'un document XML

- Un document XML est structuré en 3 parties:
 - ❖ un prologue qui contient toutes les informations autres que les données ou les éléments,
 - ❖ l'arbre des éléments avec un élément racine,
 - ❖ éventuellement des commentaires.

Prologue

- **Déclaration XML**

- Une instructions de traitement:

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

- ❖ **version** : indique que ce document est conforme à la version 1.0 de la norme XML.
- ❖ **encoding** : indique le jeu de caractères d'encodage utilisé dans le document XML.
- ❖ **Standalone** : dépendance du document par rapport à une déclaration de type de document. Standalone vaut "no" si le document XML fait référence à une DTD ou à des entités définies dans un autre fichier, vaut "yes" sinon (par défaut c'est "yes").

Prologue

- **Instruction de traitement** (*processing instruction*)

permet de faire appel à un traitement en précisant le nom de l'application et les arguments nécessaires à son exécution.

- ❖ *exemple d'une instruction de traitement:*

```
<?xml-stylesheet type="text/xsl" href="biblio.xsl"?>
```

- **Déclaration de type de document**

Cette déclaration, lorsqu'elle est présente, permet de définir la structure du document. Elle peut être de deux types, externe ou interne.

Exemple de déclaration de type de document :

- ❖ *exemple d'une déclaration de type de document:*

```
<!DOCTYPE biblio SYSTEM "biblio.dtd">
```


Commentaire

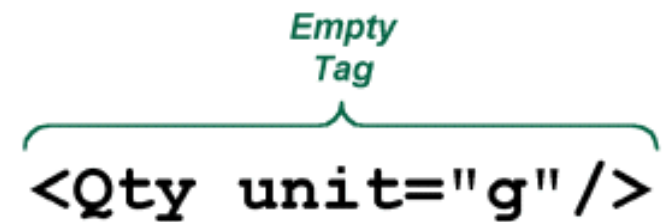
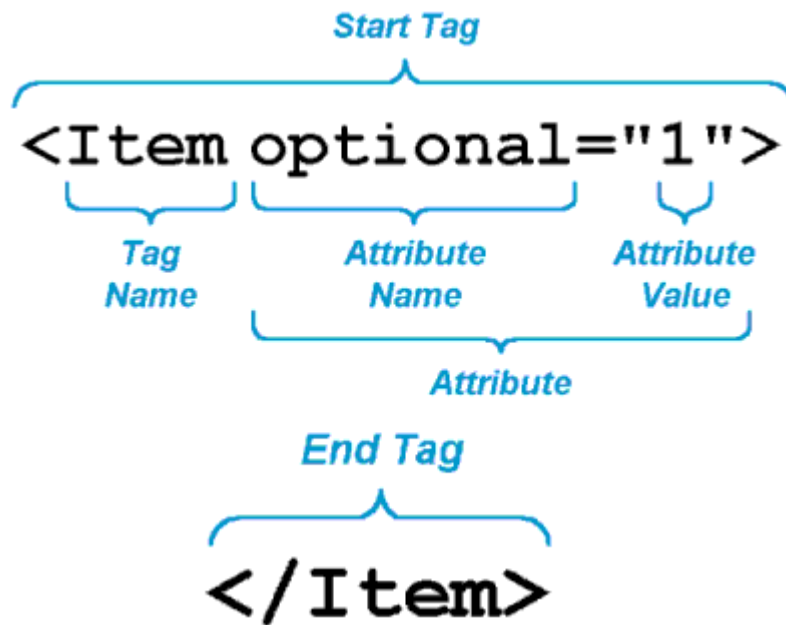
- commence donc par `<!--` et se terminent par `-->`
- peut être placés à n'importe quel endroit tant qu'ils se trouvent à l'extérieur d'une autre balise.
- Exemple:
 - ❖ `<!-- ceci est correct -->`
 - ❖ `<elt> <!-- ceci est correct aussi --> Un peu de texte </elt>`
- Remarque:
 - ❖ pas de `--` dans un commentaire.

L'arbre d'éléments

Les documents écrits avec XML ont une structure arborescente d'*éléments* dont les nœuds de l'arbre contiennent des données.

- Élément racine (*document element*)
 - ❖ est le premier élément déclaré dans un document XML, il est unique et englobe tous les autres éléments.
- Les éléments
 - ❖ forment la structure même du document
 - ❖ ce sont les branches et les feuilles de l'arborescence
 - ❖ Ils peuvent contenir du texte, ou bien d'autres éléments, qui sont alors appelés « éléments enfants ».

Balise



Exemples d'éléments

- Exemple d'élément racine:

<biblio></biblio>

- Exemple d'élément contenant du texte :

<titre>Les Misérables</titre>

- Exemple d'élément contenant d'autres éléments :

<livre>

<titre>L'Assommoir</titre>

<auteur>Émile Zola</auteur>

<couverture couleur="rouge" />

</livre>

- D'autres éléments sont vides : ils ne contiennent pas d'élément-enfant.
Exemple d'élément vide :

<couverture couleur="rouge" />

Les attributs

- Caractéristique de l'élément.
- Tous les éléments peuvent contenir un ou plusieurs attributs.
- Chaque élément ne peut contenir qu'une fois le même attribut.
- Un attribut est composé d'un nom et d'une valeur.
- Il ne peut être présent que dans la balise *ouvrante* de l'élément (par exemple, on n'a pas le droit d'écrire `</livre lang="en">`).

- Exemple d'utilisation d'un élément avec attribut :

`<instrument type="vent">trompette</instrument>`

- Exemple d'utilisation d'un élément vide avec attributs :

``

Les entités

- Les entités sont des unités de stockage.
- Il existe des entités définissables et définies. Elles peuvent être analysables ou non, internes ou externes.
- La déclaration des entités s'effectue au sein de la DTD. Elles peuvent être utilisées aussi bien dans la DTD que dans le document XML.
- L'appel d'une entité dans un document : **&nom_entite;** .
- Tous les caractères peuvent être remplacés par une entité numérique à travers un code décimal `&#code_car;` (par ex. `A` pour le A, le caractère *é* pour `é`).
- Certains caractères, lorsqu'ils sont insérés dans un document XML, doivent être remplacés par des entités prédéfinies pour qu'ils puissent être affichés (les caractères spéciaux). Ces entités prédéfinies sont :

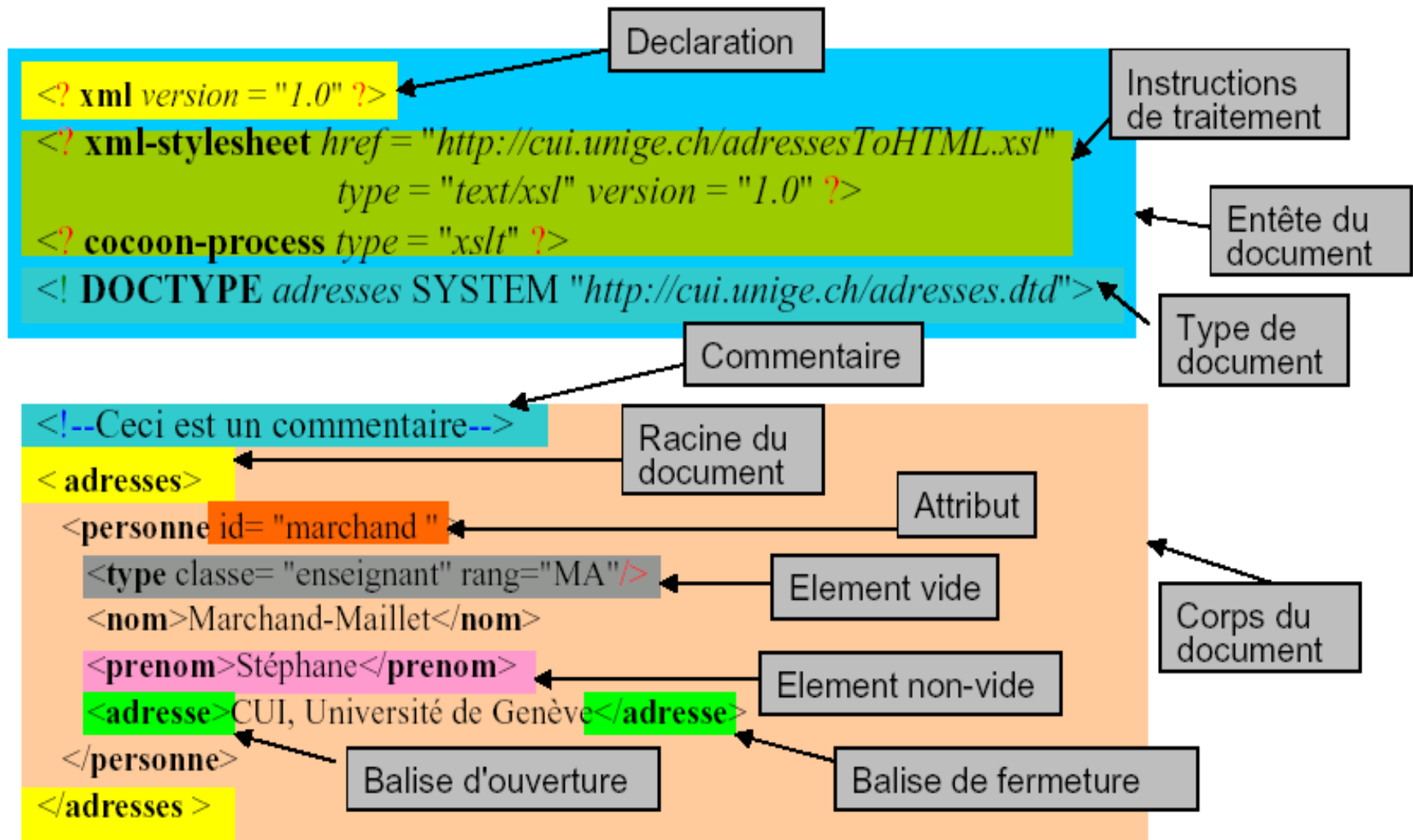
Caractère	Entité
&	&
<	<
>	>
"	"
'	'

Section CDATA

- CDATA: Character Data (données caractères)
- Est une section pouvant contenir toute sorte de chaîne de caractères.
- Le texte contenu dans la section CDATA n'est pas parsé (analysé par le processeur XML) et ses caractères de balisage ignorés.
- Ceci permet entre autres de garder dans un bloc de texte un exemple de code à afficher tel quel.
- Pas besoin de recourir à des entités pour afficher les caractères réservés de XML.
- Exemple d'utilisation de CDATA :

`<![CDATA[Une balise commence par un < et se termine par un >.]>`

Exemple complet



Règles de composition

- Un nom d'élément ne peut commencer par un chiffre.
 - ❖ Si le nom n'est composé que d'un seul caractère, ce doit être une lettre comprise entre "a" et "z" pour les minuscules, "A" et "Z" pour les majuscules.
 - ❖ S'il est composé d'au moins deux caractères, le premier peut être "_" ou ":". Le nom peut ensuite être composé de lettres, chiffres, tirets, tirets bas et deux points.
 - ❖ La syntaxe XML est sensible à la casse (le format distingue majuscules et minuscules).
- Toutes les balises portant un contenu non vide doivent être fermées.
- Les balises n'ayant pas de contenu doivent se terminer par />.
- Les valeurs d'attributs doivent être *entre guillemets*.
- Les éléments doivent être correctement imbriqués (le dernier ouvert est le premier à fermer).
- L'élément racine doit être unique.

Document « bien formé »

- Un document respectant les règles de XML citées ci-dessus est appelé **document bien formé** («*well-formed*»).
- Seuls les documents "bien formés" seront affichés correctement par un navigateur web.
- A la moindre erreur de syntaxe, le document ne sera pas ou ne sera que partiellement affiché .

Merci pour votre attention