

Chapitre 2:
Structure d'un document XML

- Introduction**
 - ✓ XML VS HTML
 - ✓ 1^{er} EXEMPLE EN XML
 - ✓ Représentation arborescente
 - ✓ Visualisation sous un navigateur
 - ✓ 1^{er} exemple en HTML
 - ✓ Représentation arborescente
 - ✓ Visualisation sous un navigateur
 - ✓ Un code html au sein d'un document XML
 - ✓ Visualisation sous un navigateur
- Structure d'un document XML**
 - ✓ Le prologue
 - ✓ La déclaration du type de document
 - ✓ Les instructions de traitement
 - ✓ Le Corps
 - ✓ Les éléments
 - ✓ Les attributs
 - ✓ Attribut ou élément, lequel choisir?
 - ✓ Les commentaires
- Règles de syntaxe**
- Parseurs XML**
- Conclusion**

INTRODUCTION

- Un document XML est un **fichier texte** contenant des données délimitées par des balises (ouvrantes et fermantes) qui permettent une forte structuration des données;
- La syntaxe du langage XML est relativement simple, elle très similaire à celle du langage HTML mais elle est plus générale (les balises ne sont pas prédéfinies) et plus stricte (il fallait respecter les règles de syntaxe).

2

XML VS HTML

- Un document HTML est un fichier texte interprétable directement par un navigateur dont les balises sont prédéfinies et il est utilisé essentiellement pour visualiser les pages Web;
- Un document XML est un fichier texte non pas interprétable directement par un navigateur dont les balises sont libres et il est utilisé dans plusieurs contextes.

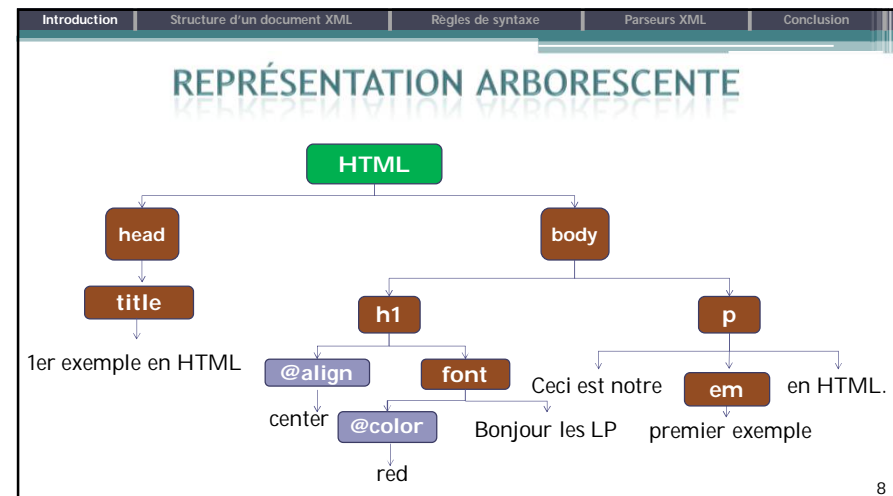
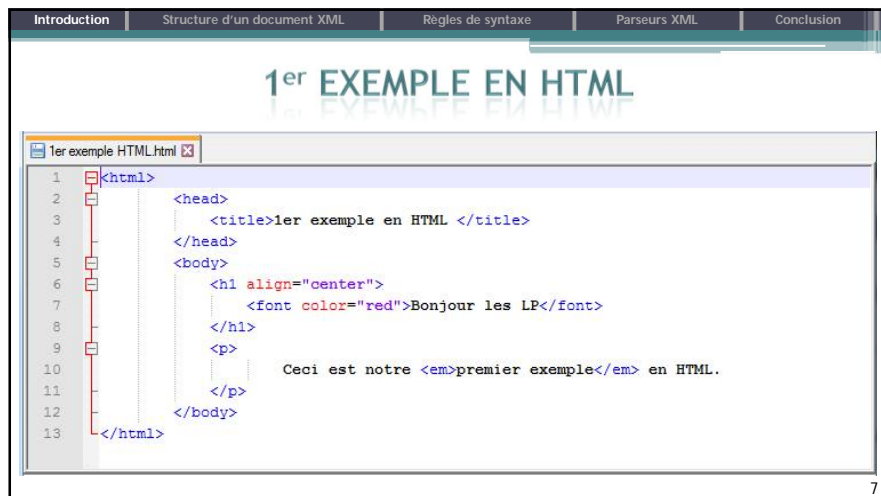
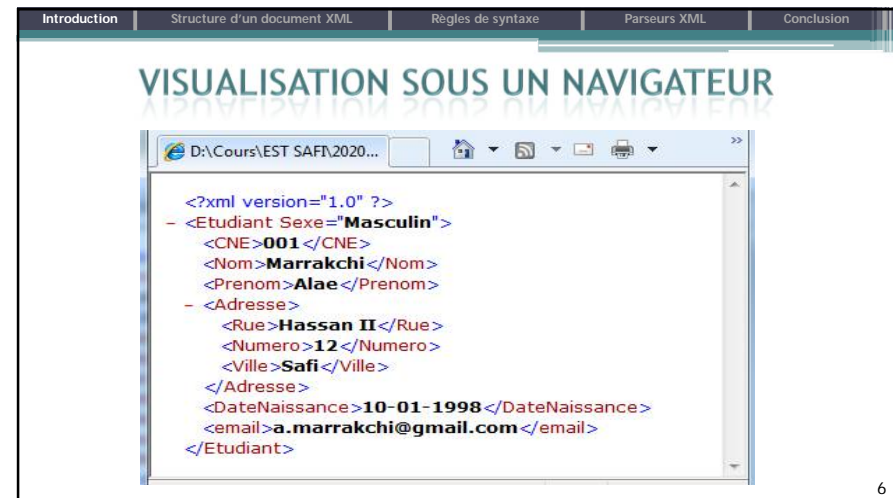
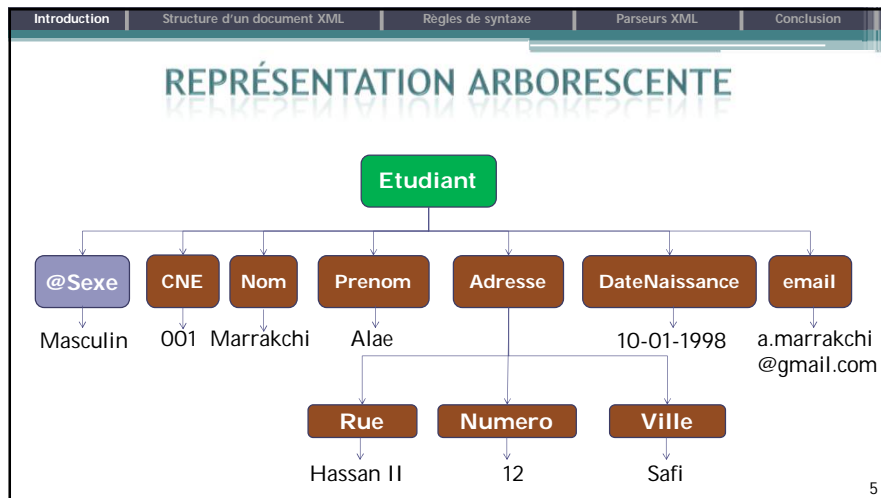
3

1^{er} EXEMPLE EN XML

```

1  <?xml version='1.0'?>
2  <Etudiant Sexe="Masculin">
3      <CNE>001</CNE>
4      <Nom>Marrakchi</Nom>
5      <Prenom>Alae</Prenom>
6      <Adresse>
7          <Rue>Hassan II</Rue>
8          <Numero>12</Numero>
9          <Ville>Safi</Ville>
10     </Adresse>
11     <DateNaissance>10-01-1998</DateNaissance>
12     <email>a.marrakchi@gmail.com</email>
13 </Etudiant>
  
```

4



Introduction Structure d'un document XML Règles de syntaxe Parseurs XML Conclusion

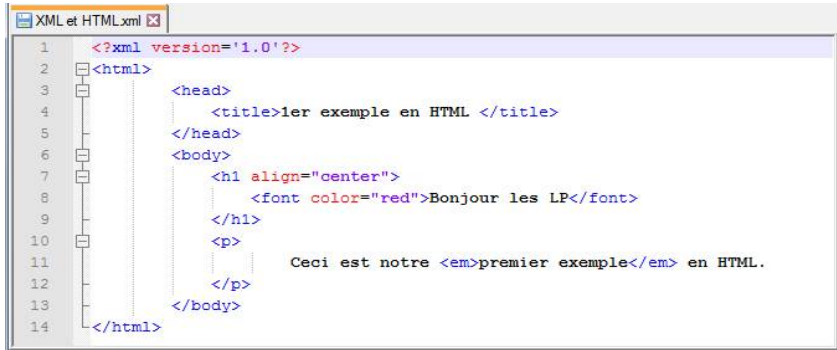
VISUALISATION SOUS UN NAVIGATEUR



9

Introduction Structure d'un document XML Règles de syntaxe Parseurs XML Conclusion

UN CODE HTML AU SEIN D'UN DOCUMENT XML



```


1 <?xml version='1.0'?>
2 <html>
3   <head>
4     <title>1er exemple en HTML </title>
5   </head>
6   <body>
7     <h1 align="center">
8       <font color="red">Bonjour les LP</font>
9     </h1>
10    <p>
11      Ceci est notre <em>premier exemple</em> en HTML.
12    </p>
13  </body>
14 </html>

```

10

Introduction Structure d'un document XML Règles de syntaxe Parseurs XML Conclusion

VISUALISATION SOUS UN NAVIGATEUR



```

<?xml version="1.0" ?>
- <html>
- <head>
-   <title>1er exemple en HTML</title>
- </head>
- <body>
-   <h1 align="center">
-     <font color="red">Bonjour les LP</font>
-   </h1>
-   <p>
-     Ceci est notre
-     <em>premier exemple</em>
-     en HTML.
-   </p>
- </body>
- </html>

```

11

Introduction Structure d'un document XML Règles de syntaxe Parseurs XML Conclusion

STRUCTURE D'UN DOCUMENT XML

- En général un document XML est composé des parties suivantes :
 - ✓ Le Prologue (appelé aussi entête);
 - ✓ La déclaration du type de document;
 - ✓ Les instructions de traitement;
 - ✓ Le corps;
 - ✓ Éventuellement des commentaires.

12

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LE PROLOGUE

- Le prologue est une déclaration (considéré parfois comme une instruction de traitement) indiquant à l'application qui va traiter le document XML les informations décrites par les attributs de la syntaxe suivante:

Syntaxe: `<?xml version=" 1.0 | 1.1" [encoding = "encodage"] [standalone = "yes | no"] ?>`

- ✓ **version:** indique la version de la norme XML (les versions actuelles: 1.0 et 1.1);
- ✓ **encoding:** attribut optionnel qui indique le codage de caractères utilisés dans le document (les principaux systèmes de codages sont: US-ASCII, ISO-8859-1, UTF-8 et UTF-16). La valeur par défaut de cet attribut est "UTF-8" ;

13

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LE PROLOGUE

- ✓ **standalone:** attribut optionnel qui indique que le document est autonome c'est-à-dire qu'il est indépendant des autres documents externe dans ce cas il prend la valeur "yes", ou bien le document dépend d'autres documents externes (DTD par exemple) dans ce cas il prend la valeur "no". La valeur par défaut de cet attribut est "no".
- Le prologue est optionnel (si le prologue n'est pas déclaré l'application utilise les valeurs par défaut des attributs encoding et standalone);
- Le prologue doit être déclaré au début du document (pas de blancs et pas de retour à la ligne avant le prologue).

14

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LA DÉCLARATION DU TYPE DE DOCUMENT

- La déclaration du type permet de définir la structure du document, c'est-à-dire les différentes règles appliquées aux éléments pour avoir un document valide, cette déclaration peut être incluse dans le même document (interne) ou être dans un document externe, elle prend généralement les deux syntaxes suivantes :

Syntaxes:

<code><!DOCTYPE racine [déclarations] ></code>	DTD interne
<code><!DOCTYPE racine SYSTEM PUBLIC "Chemin DTD URI DTD"></code>	DTD externe

- NB:** nous détaillerons les DTD dans le chapitre 3.

15

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LES INSTRUCTIONS DE TRAITEMENT

- Une instruction de traitement permet de communiquer des informations aux applications utilisant le document XML, elle est délimitée par les chaînes de caractères "<?" et "?>", elle peut être insérée n'importe où après le prologue et en dehors des balises et du contenu du DTD;
- Un exemple très fréquent d'utilisation est la transformation d'un document XML en un document XHTML par un navigateur via l'instruction suivante : `<?xml-stylesheet type="text/xsl" href="fichier.xsl"?>`
- NB:** nous détaillerons les transformations dans le chapitre 6.

16

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LE CORPS

- Le corps d'un document XML est la partie contenant les données et leurs structures, il est organisé hiérarchiquement sous forme d'un arbre composé de balises (ouvrantes et fermantes) qui représentent les **éléments** (appelés aussi **nœuds**), il y a un seul élément appelé **élément racine** qui contient tous les autres éléments;
- Un élément peut être complété par des données supplémentaires appelées **attributs** qui pourront être insérées au sein de la balise ouvrante de l'élément.

17

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LE CORPS

Un attribut

Une donnée

Une balise ouvrante

Une balise fermante

Elément

L'élément racine

```

<Etudiant Sexe="Masculin">
  <CNE>001</CNE>
  <Nom>Marrakchi</Nom>
  <Prenom>Alae</Prenom>
  <Adresse>
    <Rue>Hassan II</Rue>
    <Numero>12</Numero>
    <Ville>Safi</Ville>
  </Adresse>
  <DateNaissance>10-01-1998</DateNaissance>
  <email>a.marrakchi@gmail.com</email>
</Etudiant>
  
```

18

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LES ÉLÉMENTS

- Les éléments (appelés aussi balises ou nœuds) permettent de définir les structures de données d'un document XML, c'est-à-dire qu'ils structurent et organisent les données dans le document.
- La syntaxe générale d'un élément est la suivante :

Syntaxe: `<nomElement [attributs] >[contenu] </nomElement >`

19

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LES ÉLÉMENTS

- Généralement un élément peut avoir un contenu sous les formes suivantes :

- ✓ `<nomElement >texte</nomElement >`: il s'agit d'un élément ayant un contenu textuel (**contenu simple**); **Exemple:** `<Ville>Safi</Ville>`
- ✓ `<nomElement >` : il s'agit d'un élément (parent) qui contient d'autres éléments (fils) (**contenu complexe**)

Exemple:

```

<Adresse>
  <Rue>Hassan II</Rue>
  <Numero>12</Numero>
  <Ville>Safi</Ville>
</Adresse>
  
```

20

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LES ÉLÉMENTS

✓ **<nomElement ></nomElement >**: il s'agit d'un élément qui n'a pas de contenu (**contenu vide**), il s'écrit aussi sous la forme: **<nomElement/>**;

✓ **<nomElement >**
 du texte **<nomAutreElement >** la suite **</nomAutreElement >** encore la suite du texte.
</nomElement >

:il s'agit d'un élément qui contient un mélange des textes et des éléments (**contenu mixte**)

Exemple:
<Resume>
 Comprendre la structure **<Gras>** arborescente **</Gras>** des documents XML.
</Resume>

21

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LES ATTRIBUTS

➔ Un attribut est une donnée supplémentaire attachée à un élément, il est sous forme d'un couple (clé, valeur) qui se trouve toujours au sein de la balise ouvrante de l'élément, il prend généralement la forme suivante:

<nomElement cle1="val1" ... cleN="valN" >[contenu] </nomElement >

- ✓ Les valeurs sont délimitées soit par des guillemets « " » soit par des apostrophes « ' »;
- ✓ L'ordre des attributs au sein d'une balise n'a pas d'importance;
- ✓ Un élément peut avoir de **0** à **n** attributs distincts (ayant des clés différentes).

22

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

ATTRIBUT OU ÉLÉMENT, LEQUEL CHOISIR?

➔ Les éléments et les attributs permettent tous les deux la structuration et le stockage des données. Soit les deux exemples suivants:

Exemple 1:
<Etudiant Sexe="Masculin">
<CNE>001</CNE>
<Nom>Marrakchi</Nom>
<Prenom>Alae</Prenom>
</Etudiant>

Exemple 2:
<Etudiant >
<Sexe>Masculin</Sexe>
<CNE>001</CNE>
<Nom>Marrakchi</Nom>
<Prenom>Alae</Prenom>
</Etudiant>

➔ Dans le 1^{er} exemple la donnée sexe de l'étudiant est un attribut, tandis que dans le 2^{ème} exemple la même donnée est un élément, les deux exemples permettent de structurer les mêmes données.

23

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

ATTRIBUT OU ÉLÉMENT, LEQUEL CHOISIR?

➔ Pour faire le choix il faut prendre en considération les critères suivant :

- ✓ Un attribut d'un élément a **une clé unique** et il ne peut contenir qu'une seule valeur, tandis qu'un élément peut contenir **plusieurs éléments** y compris des éléments qui portent le **même nom** ;
- ✓ Un attribut ne peut pas contenir une structure arborescente (des textes et des balises), tandis qu'un élément peut contenir une structure arborescente ;
- ✓ L'ordre des attributs dans un élément n'a pas d'importance, tandis qu'il est possible d'imposer un ordre sur les éléments via les langages de définition de schéma;
- ✓ Un attribut n'est pas facilement extensible, tandis qu'un élément, nous pouvons facilement lui ajouter de nouveaux éléments.

24

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

ATTRIBUT OU ÉLÉMENT, LEQUEL CHOISIR?

- Il est conseillé d'utiliser les attributs dans les cas suivants :
 - ✓ Lorsqu'une donnée ne se répète pas et qui ne peut pas évoluer dans le temps ;
 - ✓ Lorsque la donnée caractérise l'élément en question ;
 - ✓ les métadonnées (données sur les données) .
- Dans tous les autres cas utiliser les éléments;
- Si vous avez du doute dans le choix entre un élément et un attribut, opter toujours pour le choix de l'élément (évolutif).

25

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

LES COMMENTAIRES

- Un commentaire est un fragment du code qui ne sera pas interprété, en XML les commentaires ont la même syntaxe qu'en HTML (XML et HTML sont dérivés du SGML) ils sont délimités par les chaînes de caractères "<!--" et "-->", ils peuvent inclure n'importe quelle chaîne de caractères sauf "--" (qui est considéré comme la fin du commentaire), ils peuvent être insérés n'importe où après le prologue et en dehors des balises.

26

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

RÈGLES DE SYNTAXE

- XML est un langage extensible, c'est-à-dire que les noms des balises ne sont pas prédéfinies (nous devons définir nos propres balises);
- Pour écrire un document XML dit **bien formé** (respecte la syntaxe XML) il faut respecter les règles suivantes :
 - ✓ Le nom d'un élément ou d'un attribut: ne doit pas commencer par un chiffre, ne doit pas contenir un espace, les caractères accentués sont acceptés mais il est préférable de les éviter car ils sont mal interprétés par certaines applications;

27

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

RÈGLES DE SYNTAXE

- ✓ Le document doit avoir un seul élément racine qui englobe tous les autres éléments;

<pre><?xml version='1.0'?> <Etudiant Sexe="Masculin"> <CNE>001</CNE> <Nom>Marrakchi</Nom> <Prenom>Alae</Prenom> </Etudiant></pre> <p>Document bien formé</p>	<pre><?xml version='1.0'?> <Etudiant Sexe="Masculin"> <CNE>001</CNE> <Nom>Marrakchi</Nom> <Prenom>Alae</Prenom> </Etudiant> <Etudiant Sexe="Féminin"> <CNE>002</CNE> <Nom>Nasih</Nom> <Prenom>Laila</Prenom> </Etudiant></pre> <p>Erreurs de syntaxe</p>
--	--

28

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

RÈGLES DE SYNTAXE

- ✓ Pour chaque balise ouvrante doit correspondre une balise fermante, sauf si l'élément est vide il lui correspond une balise auto fermante ;

<code><?xml version='1.0'?></code>	<code><?xml version='1.0'?></code>
<code><Etudiant Sexe="Masculin"></code>	<code><Etudiant Sexe="Masculin"></code>
<code><CNE>001</CNE></code>	<code><CNE>001</code>
<code><Nom>Marrakchi</Nom></code>	<code><Nom>Marrakchi</code>
<code><Prenom>Alae</Prenom></code>	<code><Prenom>Alae</code>
<code></Etudiant></code>	<code></Etudiant></code>

Document bien formé **Erreurs de syntaxe**

- ✓ Les noms des balises doivent respecter les majuscules et les minuscules (XML est sensible à la casse); `<Nom>Marrakchi</nom>` **incorrect** `<Nom>Marrakchi</Nom>` **correct**

29

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

RÈGLES DE SYNTAXE

- ✓ Les attributs d'un élément doivent être distinctes ;

<code><Matiere Note="15" Note="17"></code>	<code><Matiere NoteDS="15" NoteExam="17"></code>
XML	XML
<code></Matiere></code>	<code></Matiere></code>

incorrect **correct**

- ✓ Les valeurs des attributs doivent être délimitées par les apostrophes ou les guillemets;

<code><Matiere NoteDS=15 NoteExam=17></code>	<code><Matiere NoteDS="15" NoteExam='17'></code>
...	...

incorrect **correct**

- ✓ Les éléments doivent être correctement imbriqués.

<code><Etudiant Sexe="Masculin"></code>	<code><Etudiant Sexe="Masculin"></code>
<code><CNE>001</Etudiant></code>	<code><CNE>001</CNE></code>
<code></CNE></code>	<code></Etudiant></code>

incorrect **correct**

30

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

PARSEURS XML

➡ Il existe des outils (logiciels, applications web) qui permettent de vérifier la syntaxe d'un document (bien formé ou non) et d'autres fonctionnalités, ils sont appelés des parseurs ou des analyseurs, le tableau suivant illustre quelques logiciels:

Logiciel	Téléchargement	Licence	Plateforme
XML copy editor	http://xml-copy-editor.sourceforge.net/	logiciel libre sous licence GNU General Public License.	Linux Windows MacOS
Oxygen	http://www.oxygenxml.com/	logiciel propriétaire qui propose une version gratuite pendant 30 jours.	
EditIX	http://www.editix.com/	logiciel propriétaire qui propose une version libre plus légère.	

31

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

PARSEURS XML

➡ Nous vous proposons à titre d'exemple les applications web suivantes:

- ✓ <http://www.xmlvalidation.com/> (consulté le 03/12/2020): il permet d'analyser un document à partir d'un URL, copier le contenu du document dans une zone de texte puis l'analyser ou bien télécharger le document vers le serveur de l'application;
- ✓ <http://validator.aborla.net/> (consulté le 03/12/2020): il permet d'analyser un document à partir d'un URL, ou bien télécharger le document vers le serveur de l'application;

➡ Nous pouvons aussi utiliser la commande `xmllint` dans un terminal Linux.

32

Introduction | Structure d'un document XML | Règles de syntaxe | Parseurs XML | Conclusion

CONCLUSION

- XML est un langage de balisage extensible dont les balises ne sont pas prédéfinies;
- Un document XML est organisé sous forme d'un arbre ayant un seul nœud dit élément racine qui englobe tous les autres nœuds de l'arbre;
- Les données sont structurées en éléments qui sont qualifiés par des attributs;
- C'est un langage strict en terme de syntaxe;
- Un document qui respecte la syntaxe XML est un document **bien formé**.

33

Merci pour votre attention !