XML(Extensible Markup Language)

- XML est un format/langage standard conçu pour le stockage et la publication et l'échange de documents entre sites distants ou applications différentes.
- XML n'a pas de balises prédéfinies et permet aux créateurs de spécifier leur propre jeu de balise pour structurer leurs données
- Un métalangage (permet de définir d'autres langages)

```
▼<itineraire>
    <etape distance="0km">départ</etape>
    <etape distance="13km">tourner à droite</etape>
    <etape distance="22km">arrivée</etape>
  </itineraire>
```

Technologies liées à XML

- **XSL**, langage évolué pour la définition de feuilles de style.
- **Xlink** pour ajouter des liens hypertextes à un fichier XML.
- **XPath** pour identifier des nœuds ou ensemble de nœuds dans un document XML
- **Xquery** langage de requête permettant d'extraire des informations d'un document XML
- **DOM** Document Object Model pour construire en mémoire le document XML sous forme arborescente
- Etc.

Un document XML bien formé

Un document XML bien formé est un document qui est conforme aux règles syntaxiques du langage XML. Les règles que doit suivre un document XML sont les suivantes:

1. Commence par une déclaration XML (attribut **version** obligatoire) avec possibilité de choisir un encodage (le défaut est utf-8): `<?xml version="1.0" encoding="ISO-8859-1"?>`
2. Structure hiérarchique:

- veiller à l'ordre de fermeture des balises : la première ouverte est toujours la dernière fermée, pas de croisements de type `<i>...<b>...</i> .... </b>`
- sensible à la casse

- les valeurs des attributs sont entre guillemets ou apostrophes ;
- un seul élément racine (root):
- l'élément root ne peut apparaître qu'une fois et ne doit pas apparaître dans un autre élément
- les caractères réservés sont remplacés par des références d'entités: `<, &, >," ’`

  - utilisez `< & > &aquot; &apos; …`

Un document est valide s'il :

- Est bien formé,
- Fait référence à une DTD (Document Type Definition) ou à un schéma XSD,
- Se conforme à la DTD ou au schéma XSD.

Structure d'un document XML

Un document XML comporte des éléments avec ou sans attributs qui fournissent des méta-informations sur l'information ou sur le contenu du document. Un document XML comporte :

- Un **prologue** qui contient toutes les informations autres que les données ou les éléments,
- L'arbre des éléments avec un élément racine,
- Des commentaires.
- Des instructions de traitement

1. Prologue:

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

- *version* : de façon générale *1.0*
  - Il existe deux versions 1.0 et 1.1
- *encoding* : valeur par défaut *UTF-8*
- *standalone* : valeur par défaut *yes*
  - Cette information permet de savoir si votre document XML est autonome ou si un autre document lui est rattaché.

Commentaires XML: On peut placer des commentaires à peu près partout dans un document XML. La syntaxe est identique à celle d'un fichier HTML. Un commentaire peut d'étendre sur plusieurs lignes. La seule contrainte est de ne pas pouvoir employer les caractères -- dans le commentaire, même s'ils ne sont pas suivis de >

Les instructions de traitement: sont destinées aux applications qui traitent les documents XML. Les instructions de traitement sont délimitées par les chaînes de caractères '<?' et '?>'.

```
<?nom arg1 arg2 ... argn ?>
```

- Appel d'un processeur XSLT

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

Espaces de nom: Les espaces de noms (ou namespace) sont destinés à lever les ambiguïtés éventuelles des intitulés de balise, au moyen d'un identifiant de ressource uniforme (URI)

- On utilise le symbole « **:** » dans les noms XML et tout ce qui précède le deux-points est appelé le « **préfixe** ».

- Tout attribut ou élément qui utilise un **préfixe** donné fait automatiquement partie de l'espace de noms identifié par l'URI.
- On définit un « **préfixe** » à l'aide d'un attribut ayant comme préfixe « **xmlns** ». La définition du préfixe est alors valable pour l'ensemble de l'élément

```
<root>
<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

<f:table xmlns:f="https://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>

</root>
```

```
<root xmlns:h="http://www.w3.org/TR/html4/"
xmlns:f="https://www.w3schools.com/furniture">

<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>

</root>
```

Espaces de noms par défaut

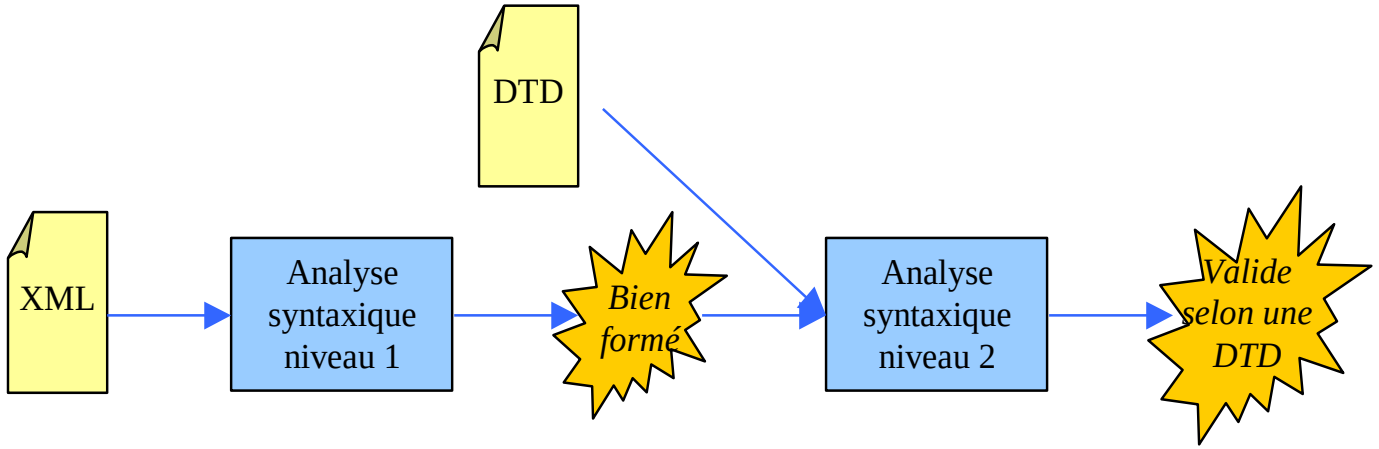
- La déclaration d'un espace de noms par défaut se fait dans le premier élément qui utilise le vocabulaire, grâce au mot clef **xmlns** comme **XML** namespace

```
<table xmlns="https://www.w3schools.com/furniture">
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- La ligne ci-dessus est le prologue -->
<!-- une instruction de traitement -->
<?xml-stylesheet type="text/xsl" href="MyXsl.xsl"?>
<!-- Élément racine -->
<biblio>
  <!-- Premier enfant -->
  <livre>
    <!-- Élément enfant titre -->
    <titre>Les Mis&#233;rables</titre>
    <auteur>Victor Hugo</auteur>
    <nb_tomes>3</nb_tomes>
    <prix>prix &lt; 50.00$ </prix>
  </livre>
  <livre>
    <titre>L&apos;Assomoir</titre>
    <auteur>&#233;mile Zola</auteur>
  </livre>
  <livre lang="en">
    <titre>David Copperfield</titre>
    <auteur>Charles Dickens</auteur>
    <nb_tomes>3</nb_tomes>
  </livre>
</biblio>
```

Validation de document avec DTD (Document Type Definition) :

- Afin de vérifier qu'un document XML est conforme à une syntaxe donnée, DTD nous permet de décrire la structure et l'ensemble des règles que doit suivre le document:
  - liste des balises
  - attributs des balises
  - organisation des balises
- La validation est le mécanisme qui vérifie qu'un document XML respecte une DTD



- On distingue 2 types de DTD : les *internes* et les *externes*.

- Une **DTD** *interne* s'écrit dans ce qu'on appelle le **DOCTYPE**. On le place sous le prologue du document et au dessus du contenu XML. Voyons plus précisément la syntaxe :

```
<!DOCTYPE racine [ ]>
```

La **DTD** *interne* est ensuite écrite entre les []. Dans ce DOCTYPE, le mot *racine* doit être remplacé par le nom de la balise qui forme la racine du document XML.

- Une **DTD** *externe* est une **DTD** qui est écrite dans un autre document que le document XML. Il existe 2 types de DTD : les DTD externes **PUBLIC** et les DTD externes **SYSTEM**. Avec les DTD externes **SYSTEM**, on fait appelle au fichier contenant la grammaire à partir d'un fichier local. Avec **PUBLIC,** en y accédant par son URL.

Exemple d'un document XML ayant une DTD externe *cours.dtd*

```
<?xml version="1.0"?>
<!DOCTYPE cours SYSTEM "cours.dtd">
<cours>
…
</cours>
```

Les éléments dans la DTD: pour déclarer les éléments autorisés à apparaître dans le document, ainsi que leurs imbrications possibles, on utilise la forme suivante:

```
<!ELEMENT nom (contenu)>
```

- **ELEMENT** : mot clé pour déclarer une balise
- **nom** : le nom de la balise
- (**contenu**) : représente soit un type de donnée prédéfini, soit une règle d'utilisation de l'élément.

Type	DTD	XML
Elément vide	<!ELEMENT elt *EMPTY*>	<elt/>
Elément contenant du texte	<!ELEMENT elt (*#PCDATA*)>	<elt>texte</elt>
Elément avec sous éléments	<!ELEMENT elt (sous-elt)> <!ELEMENT sous-elt *EMPTY*>	<elt> <sous-elt/> </elt>

Type	DTD	XML
Elément avec plusieurs sous éléments	*<!ELEMENT elt (s1, s2)>* *<!ELEMENT s1 EMPTY>* *<!ELEMENT s2 EMPTY>*	*<elt>* *<s1/>* *<s2/>* *</elt>*
Elément avec contenu variable	*<!ELEMENT elt (#PCDATA|s1)>* *<!ELEMENT s1 EMPTY>*	*<elt>texte</elt>* *ou* *<elt><s1/></elt>*
Elément à contenu non défini	*<!ELEMENT elt ANY>* *<!ELEMENT s1 EMPTY>*	*<elt>texte</elt>* *ou* *<elt><s1/></elt>* *etc.*

Il est possible de moduler le nombre d'apparitions d'un sous-élément en utilisant des quantifieurs après les noms d'éléments:

Cardinalité	Signification
<!ELEMENT elt (s1)>	Le contenu de la balise elt est une seule balise s1
<!ELEMENT elt (s1*)>	Le contenu de la balise elt est 0 à n occurrence de la balise s1
<!ELEMENT elt (s1?)>	Le contenu de la balise elt est 0 à 1 occurrence de la balise s1
<!ELEMENT elt (s1+)>	Le contenu de la balise elt est 1 à n occurrence de la balise s1

Quelques exemples :

```
<!ELEMENT plan (introduction?,chapitre+,conclusion?)>
```

L'élément `plan` contient un élément `introduction` optionnel, suivi d'au moins un élément `chapitre` et se termine par un élément `conclusion` optionnel également.

```
<!ELEMENT chapitre (auteur*,paragraphe+)>
```

L'élément `chapitre` contient de 0 à n éléments `auteur` suivi d'au moins un élément `paragraphe`.

Prenons l'exemple suivant:

```
<!ELEMENT personne (nom, prenom?)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
```

Cette définition impose que la balise **<personne/>** contienne obligatoirement la balise **<nom />** puis éventuellement **<prenom/>**

Regardons alors la validité de ces documents XML :

```xml
<!-- valide -->
<personne>
    <nom>DOE</nom>
</personne>

<!-- valide -->
<personne>
    <nom>DOE</nom>
    <prenom>John</prenom>
</personne>

<!-- invalide -->
<!-- l'ordre des balises n'est pas respecté -->
<personne>
    <prenom>John</prenom>
    <nom>DOE</nom>
</personne>
```

Déclaration des attributs: Les attributs sont précisés dans l'instruction **ATTLIST**. Cette dernière, étant indépendante de l'instruction **ELEMENT**, on précise à nouveau le nom de l'élément sur lequel s'applique l'attribut. Une déclaration d'attributs typique aura la forme suivante:

```
<!ATTLIST nom nom-attribut type contrainte "valeur-par-defaut">
```

- **ATTLIST** : mot clé pour déclarer un attribut
- *nom* : le nom de la balise
- *nom-attribut* : le nom de l'attribut
- *type* : type de l'attribut
  - CDATA : chaîne de caractères
  - liste de valeurs possibles `("v1"|"v2"|"v3")`
- *valeur-par-defaut*: valeur par défaut de l'attribut si aucune valeur n'est fournie

- Contrainte : définit les contraintes que doit respecter l'attribut, n'est pas obligatoire.
  - REQUIRED : l'attribut est obligatoire
  - IMPLIED : l'attribut est optionnel
  - FIXED : la valeur de l'attribut est fixé
- Un attribut peut être déclaré comme identifiant unique d'un élément avec le mot clé **ID.** Un attribut peut être déclaré comme référence vers un autre élément (clé étrangère) avec le mot clé **IDREF**

```
<! ATTLIST disque IDdisk ID #REQUIRED type(K7|MiniDisc|Vinyl|CD)"CD" >
```

Ce qui signifie que l'on affecte à l'élément ***disque*** deux attributs *IDdisk* et *type*. Le premier attribut est un identifiant unique obligatoire. L'attribut *type* peut être soit K7, MiniDisc, Vinyl ou CD, sachant que ce dernier sera affecté par défaut.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ecrivain (#PCDATA)>
<!ELEMENT bibliotheque (ecrivain*,livre*)>
<!ELEMENT livre (auteur)>
<!ELEMENT auteur EMPTY>
<!ATTLIST ecrivain identifiant ID #REQUIRED>
<!ATTLIST auteur identifiant IDREF #REQUIRED>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bibliotheque SYSTEM "parent.dtd" >
<bibliotheque>
    <ecrivain identifiant="A">E1</ecrivain>
    <ecrivain identifiant="B">E2</ecrivain>
    <livre>
        <auteur identifiant="A"></auteur>
    </livre>
    <livre>
        <auteur identifiant="B"></auteur>
    </livre>
    <livre>
        <auteur identifiant="A"></auteur>
    </livre>
</bibliotheque>
```

```
<?xml version="1.0" encoding="utf-8"?>
  <!DOCTYPE itineraire [
    <!ELEMENT itineraire (etape+)>
    <!ATTLIST itineraire nom CDATA #IMPLIED>
    <!ELEMENT etape (#PCDATA)>
    <!ATTLIST etape distance CDATA #REQUIRED>
  ]>

<itineraire nom="essai">
    <etape distance="0km">départ</etape>
    <etape distance="1km">tourner à droite</etape>
</itineraire>
```

```
<?xml version="1.0" encoding="utf-8"?>
  <!DOCTYPE itineraire [
    <!ELEMENT itineraire (boucle?, etape+, variante*)>
    <!ELEMENT boucle EMPTY>
    <!ELEMENT etape (#PCDATA)>
    <!ELEMENT variante ANY>
  ]>
<itineraire>
    <boucle/>
      <etape>départ</etape>
      <etape>tourner à droite</etape>
      <variante>
        <etape>départ</etape>
        <tape>tourner à gauche</tape>
      </variante>
</itineraire>
```

```
<?xml version="1.0" encoding="UTF-8"?>
    <email>
        <from> luca.rossi.917@gmail.com </from>
        <to> atzeni@dia.uniroma3.it </to>
        <content>
                Dear <person> Paolo </person>,
                here are some very hard exercises for the upcoming assignment of <course> Basi di Dati 2 </course>:
                <exercises>
                    <exercise>
                        <topic> DTD </topic>
                        <description> From Instance to DTD </description>
                    </exercise>
                    <exercise>
                        <topic> XPath </topic>
                        <description> Find students with average grade better than 26 </description>
                    </exercise>
                </exercises>
                Best Regards,
                <person> Luca </person>
        </content>
    </email>
```

```
<!DOCTYPE email [
<!ELEMENT email (from, to, content)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT content (#PCDATA|person|exercises|course)*>
<!ELEMENT exercises (exercise*)>
<!ELEMENT exercise (topic, description)>
<!ELEMENT topic (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT person (#PCDATA)>
<!ELEMENT course (#PCDATA)>
]>
```