

Le plan

Xml

- Prologue
- Commentaires
- Élément

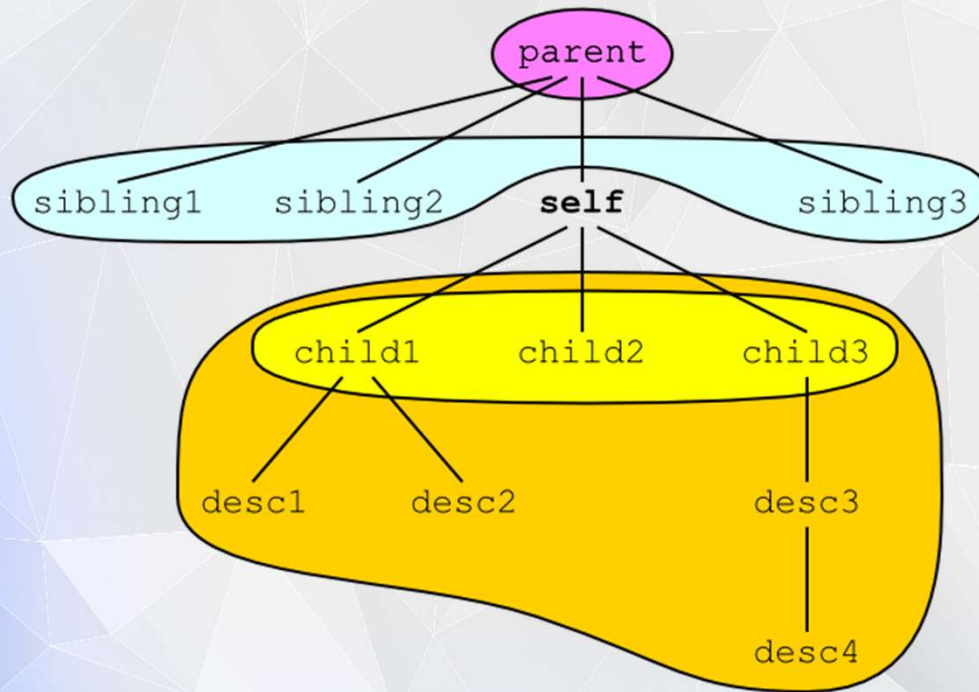
Structures

- Hiérarchique avec des nœuds
- Entités
- CDATA

Traitements

- Instruction de traitement
- URI, URN et URL
- Espaces de nommage

Notion de hiérarchie



Concept	Traduction
parent	Nœud "père"
sibling	Nœud "frère"
self	Nœud "courant"
child	Nœud "fils"
descendent (desc)	Nœud "descendant"

Nœud texte

- C'est l'information "textuelle" d'un élément

```
<personnage>  
  <nom>Mr Fernand de Montauban</nom>  
</personnage>
```

- Peut être écrit sur plusieurs lignes

```
<personnage>  
  <nom>Mr Fernand  
    de Montauban</nom>  
</personnage>
```

- Attention aux caractères spéciaux :



Les entités

Les entités sont des fragments de document XML.

Une entité peut-être considérée comme un alias permettant de réutiliser des informations au sein du document XML.

A l'origine, la DTD était utilisée pour définir les règles de construction du fichier XML. Cela a été repris par le schéma Xml.

Certains caractères sont interdits dans le contenu des éléments. Comme il est interdit de les employer, XML propose une représentation appelée *référence d'entité* ou ***entité*** pour simplifier.

Les entités

Elles sont placées:

- **après le "prologue XML"**, et **délimitées** par **<!-->**
- et sont définies dans le nœud **DOCTYPE** appelé **Document Type Definitions** concernant la racine du document.

Les entités regroupent :

- **les entités prédéfinies** (exemple : les caractères spéciaux)
- **les entités définissables** (celles définies dans le document)

Déclarer une entité

```
<!ENTITY  nom_entité  " texte_de_replacement  
ou code_de_l_entité " >
```

Appeler une entité dans le code : **&entitéDéfinie-ou-codeEntité;**

Caractères spéciaux

liste des 5 entités les plus fréquentes
prédéfinies de XML:

Nom	Caractère	Codage Unicode (décimal)	Standard	Description
quot	"	U+0022 (34)	XML 1.0	Guillemet droit
amp	&	U+0026 (38)	XML 1.0	Esperluette (« et » commercial)
apos	'	U+0027 (39)	XML 1.0	Apostrophe
lt	<	U+003C (60)	XML 1.0	Signe inférieur à
gt	>	U+003E (62)	XML 1.0	Signe supérieur à

&; affichera le caractère **&**

Les entités définies dans le document

Il est possible de créer ses propres entités.

Dans ce cas, elles peuvent représenter beaucoup plus qu'un caractère, toute une chaîne, voire même tout un arbre XML. On distingue deux «types» d'entité.

On parle **d'entité interne** (→ texte à mettre à la place)
ou **d'entité externe** (→ Il s'agit d'une entité qui appelle un autre
fichier, qui peut être un document XML ou pas)

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE personnels [                               (→ personnels = racine du document)
  <!ENTITY c-cedille "&#231;">                        (→ caractère spécial)
  <!ENTITY auteur "Robin Fran&c-cedille;oise">        (→ entité interne)
  <!ENTITY autreSection SYSTEM "autresection.xml">    (→ entité externe)
]>

<personnels>
  <personne>Durand Fran&c-cedille;oise</personne>
  <personne>&auteur;</personne>
  &autreSection;
</personnels>
```

Section "littérale" CDATA

Ce sont les données "textuelles" d'un élément, sans remplacement, contient du texte brut qui doit être inclus, mais pas analysées.

Se place dans l'élément racine. *Vous pouvez l'inclure dans un élément* `<information> « cdata » </information>`.

`<![CDATA[` Contenu textuel brut pouvant contenir des caractères spéciaux `<, > et &` `]]>`

↓ Ouverture

↓ Fermeture

Notes :

- la section littérale "CDATA" **ne peut pas contenir "**`]]>`**"** puisque c'est la chaîne de fermeture.
- On **ne peut pas imbriquer** les portions CDATA
- Le **texte** dans un CDATA **n'est pas interpréter**



SUJET:

Dans le fichier Xml du stage!

La structure Xml à respecter est la suivant:

- Un <stage> contient une liste de <stagiaires>.
- Un <stagiaire> est identifié par son <nom>, son <prenom> et son <grade>.
- Un <stagiaire> possède l'attribut armee, qui pourra avoir comme valeur "AA" ou "AT".



SUJET:

Dans le fichier Xml de la liste des chasseurs de l'AA

Ajouter les éléments suivants:

- Définir les entités :
 - **MIR** qui prendra la valeur **"Mirage"**
 - **RAF** qui prendra la valeur **"Rafale"**
 - **Remplacer les « et » par un & en passant par son code d'entité**
- Définir un contenu à afficher tel quel (CDATA):
 - la valeur : *"Pappy Boyington was born on December 4, 1912, in Coeur d'Alene, Idaho. In 1936, he enlisted in the U.S. Marine Corps and became a pilot. After Pearl Harbor, he organized the famous Black Sheep Squadron. He shot down 28 Japanese planes before he himself was shot down in 1944. He was kept as a prisoner of war until 1945. The U.S. government awarded him a Medal of Honor. He died in 1988."*

Le plan

Xml

- Prologue
- Commentaires
- Élément

Structures

- Hiérarchique avec des nœuds
- Entités
- CDATA

Traitements

- Instruction de traitement
- URI, URN et URL
- Espaces de nommage

Instruction de traitement

Dans un fichier XML, il est parfois nécessaire de noter des instructions spéciales pour traiter des informations par le logiciel de lecture.

On appelle ces instructions : instructions de traitement.

processing instructions 

Une instruction de traitement permet :

- de **transmettre le type d'application**
- et éventuellement **différents paramètres** permettant de traiter une entité externe non-analysable

comme une vidéo (avi, mpeg, qtw, etc.), un document textuel (pdf, doc, rtf, etc.) un programme Java, Perl ou encore C++.

Dans ce cours, nous l'utiliserons que pour le XSLT (transformation d'un fichier Xml) et pour le fichier .css.

Instruction de traitement

Syntaxe **<?cible liste_attributs?>**

Ouverture Fermeture

cible : identifie l'application à laquelle l'instruction est destinée.

liste_attributs : permet de spécifier des données particulières pour le traitement de l'instruction.

Les instructions de traitement débutant par la chaîne de caractères xml sont réservés à l'usage du standard XML.

Doit posséder un nom, peut contenir des attributs, peut contenir une instruction quelconque

Exemple: quand on utilisera XSLT pour transformer un flux XML

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

Qu'est ce qu'un URI ?

(UNIFORM RESOURCE IDENTIFIER) 

Les **URI** sont des liens hypertextes d'accès à une ressource distante.

La syntaxe d'un URI est la suivante :

<typedeconnexion>://<adresseserveur>/<dossier>/etc.

Par exemple : <http://www.w3.org/REC-html40>.

Les **URLs** et **URNs** sont des **spécialisations d'URI** qui permettent d'identifier une ressource soit par sa localisation, soit par son nom ou même les deux (en général l'URL contient le nom de la ressource et donc l'URN).

URL et URN ?

Les **URLs** (Uniform Resource Locators) sont les identifiants les plus connus et permettent d'identifier la localisation d'une ressource.

A partir d'une URL, je peux aller chercher la ressource correspondante car j'ai son adresse.

URI { **URL** : <http://www.ent-etrans.net/accueil>
URN : urn:etrans:cursus:diplome-technique
NID : name identifier **NSS** : name specific string

Les **URNs** (Uniform Resource Names) sont aussi des identifiants, mais ils ne permettent pas d'identifier la localisation d'une ressource.

- le numéro ISBN d'un livre
- Le numéro de sécurité sociale

Espace de nommage XML

( Namespaces)

- ☐ permet de "catégoriser" les éléments (et les attributs)
- ☐ se déclare sur un élément
 - ☐ Sur l'élément racine pour tout le document
 - ☐ Sur un élément pour cet élément et ses descendants
- ☐ un élément peut contenir la déclaration de plusieurs espaces de nommages
- ☐ un élément peut avoir un espace de nommage par défaut

Définition W3C :

Les espaces de nommage XML offrent une méthode simple pour qualifier les **noms des éléments** et des **attributs** utilisés dans des documents XML, en associant ceux-ci avec des espaces de nommage désignés par des références d'URI.

Exemple "espace de nommage"

Il est habituel d'associer l'espace de nommage XHTML au préfixe html.

```
<html:doctype  
  xmlns:html="http://www.w3.org/1999/xhtml">  
  
  <html:head>  
    <html:title>Espaces de noms</html:title>  
  </html:head>  
  <html:body>  
    ...  
  
  </html:body>  
</html:doctype>
```

Exemple

"espace de nommage par défaut"

L'espace de noms par défaut:

- **xmlns** dont la valeur est l'URI de l'espace de noms.

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <!-- L'espace de noms par défaut est celui de xhtml -->
  <head> <title>Espaces de noms</title> </head>
  <body>
    ...
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <!-- dans cet élément math et ses descendants
      L'espace de noms par défaut est maintenant celui de MathML -->
      <balises> ... </balises>
    </math>
    <!-- L'espace de noms par défaut est à nouveau celui de xhtml -->
    ...
  </body>
</html>
```

Xml

Prologue

```
<?xml version="1.0" encoding="utf-8" ?>
```

Commentaires

```
<!-- texte commentaire -->
```

Élément/ balise

```
<balise flingueur=" oui"> tonton</balise>
```

Structures

Hiérarchique avec des
nœuds

```
&amp; &quot; &apos; &lt; &gt;
```

Entités

```
<!ENTITY c-cedille "&#231;";>
```

CDATA

```
<![CDATA[Contenu avec des caractères spéciaux <, >  
et & ]]>
```

traitement

Instruction de trt

Transformation avec xslt:

URI, URN et URL

```
<?xml-stylesheet href="list.xsl" type="text/xsl"  
title="En liste" ?>
```

Espaces de nommage

SUJET: **Dans le fichier Xml de la liste des** **chasseurs de l'AA**



- ☐ Mettre en place un espace de nommage dont le préfixe sera aae, pour armée de l'air et de l'espace
 - ☐ Déclarer et utiliser dans toutes les balises
- ☐ Mettre en forme le xml avec du CSS ! (pour les confirmés)

FIN
Avez-vous des questions ?

