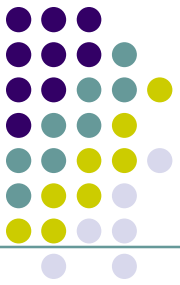


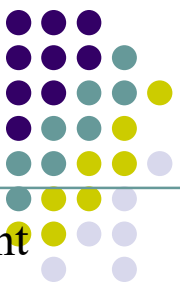


Université Gaston Berger de Saint-Louis
Année académique 2016-2017



DTD

Document Type Definition



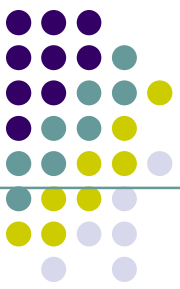
Références

1. XML Programming Success in a Day: Beginner's Guide to Fast, Easy, and Efficient Learning of XML Programming. Sam Key, 2015.
2. XML Programming: The Ultimate Guide to Fast, Easy, and Efficient Learning of XML Programming. Christopher Right, 2015
3. Beginning XML. Joe Fawcett and Danny Ayers, 2012.
4. XML, Cours et exercices. Modélisation, Schémas et DTD, design patterns, XSLT, DOM, Relax NG, XPath, SOAP, XQuery, XSL-FO, SVG, eXist. *Alexandre Brillant*, Édition : Eyrolles 2^eédition, 2010
5. Schémas XML, Jean-Jacques Thomasson, Edition Eyrolles 2002
6. World Wide Web Consortium (W3C) : w3.org
7. World Wide Web School : w3schools.com/xml
8. Xml.developpez.com
9. Youtube.com



Sommaire

- Introduction
- Les composants
- La déclaration des composants
- Déclaration des éléments
- Déclaration des attributs
- Déclaration des entités



Introduction

Qu'est ce qu'un DTD?

- DTD signifie Document Type Definition.
- Il permet de définir la structure et les éléments et attributs légaux dans un document XML.

Pourquoi utiliser une DTD?

- Pour permettre à des groupes de personnes indépendants de convenir des éléments et attributs légaux pour l'échange de données.
- Pour vérifier que les données XML sont valides.



Introduction

Déclaration d'une DTD interne

Si une DTD est déclarée dans le fichier XML, elle doit être enveloppée dans la définition `<!DOCTYPE>`:

exemple:

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (a,de,titre,corps)>
<!ELEMENT a (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT corps (#PCDATA)>
]>
<note>
<a>Jéen</a>
<de>Adama</de>
<titre>Reminder</titre>
<corps>Don't forget the course</corps>
</note>
```

Quand le fichier XML est affiché dans un navigateur, on sélectionne "voir source" pour afficher la DTD



Introduction

Déclaration d'une DTD interne

La DTD ci-dessus est interprétée comme suit :

- **!DOCTYPE note** définit que l'élément racine du document est **note**
- **!ELEMENT note** définit que l'élément **note** doit contenir quatre éléments: "a, de, titre, corps"
- **!ELEMENT a** définit l'élément **a** de type "#PCDATA"
- **!ELEMENT de** définit l'élément **de** de type "#PCDATA"
- **!ELEMENT titre** définit l'élément **titre** de type "#PCDATA"
- **!ELEMENT corps** définit l'élément **corps** de type "#PCDATA"



Introduction

Déclaration d'une DTD externe

Si la DTD est déclarée dans un fichier externe, la définition `<!DOCTYPE>` doit contenir une référence au fichier DTD:

Exemple:

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
<a>Jéen</a>
<de>Adama</de>
<titre>Reminder</titre>
<corps>Don't forget the course</corps>
</note>
```

Voici le fichier "note.dtd",
qui contient la DTD::

```
<!ELEMENT note
(a,de,titre,corps)>
<!ELEMENT a (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT corps (#PCDATA)>
```



Les composants

Du point de vue du DTD, les documents XML sont constitués par les composants suivants:

- Elements
- Attributs
- Entités
- PCDATA
- CDATA



Les composants

Elements

- Les éléments constituent les principaux composants des documents XML.
- Des exemples d'éléments XML peuvent être "note" et "message". Les éléments peuvent contenir du texte, d'autres éléments ou être vides.

Exemples:

`<corps>du texte</corps>`

`<message>du texte</message>`



Les composants

Les attributs

- Les attributs fournissent **des informations supplémentaires sur les éléments** .
- Les attributs sont toujours placés dans la balise de début d'un élément. Les attributs sont toujours en paires nom/valeur.

Exemple: ``

Le nom de l'élément est "img". Le nom de l'attribut est "src". La valeur de l'attribut est "computer.gif". Puisque l'élément lui-même est vide, il est fermé par un "/".

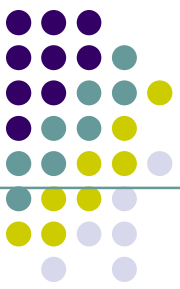


Les composants

Les entités

- Certains caractères ont une signification particulière en XML, comme le signe (<) qui définit le début d'une balise XML.
- On connaît l'entité HTML: " " utilisé pour insérer un espace supplémentaire dans un document. Les entités sont remplacées lorsqu'un document est analysé par un parser XML.
- Les entités suivantes sont prédéfinies en XML:

Référence d'entité	Caractère
<	<
>	>
&	&
"	"
'	'



Les composants

PCDATA

- PCDATA signifie données de caractères parsés.
- Pensez aux données de caractère comme le texte trouvé entre la balise de début et la balise de fin d'un élément XML.
- **PCDATA est un texte qui sera analysé par un parser. Le texte sera examiné par l'analyseur pour les entités et le balisage .**
- Les balises dans le texte seront traitées comme balises et les entités seront développées..
- Toutefois, les données de caractères analysées ne doivent pas contenir de caractères &, < ou >; Ceux-ci doivent être représentés respectivement par les entités & < et >



Les composants

CDATA

- CDATA signifie données de caractères..
- **CDATA est un texte qui ne sera PAS traité par un analyseur** . Les balises dans le texte ne seront PAS traitées comme balises et les entités ne seront pas développées.



Déclaration d'un élément

Dans une DTD, les éléments sont déclarés avec le mot ELEMENT.

Exemple:

```
<!ELEMENT nom-element categorie>
```

ou

```
<!ELEMENT nom-element (contenu de l'element)>
```

• Éléments vides

Les éléments vides sont déclarés avec le mot-clé EMPTY :

```
<!ELEMENT nom-element EMPTY>
```

Exemple:

```
<!ELEMENT br EMPTY>
```

Exemple XML :

```
<br />
```



Déclaration d'un élément

- **Éléments avec Parsed Character Data (PCDATA)**

Les éléments avec des données de caractères parsés sont déclarés avec #PCDATA entre parenthèses:

`<!ELEMENT nom-element (#PCDATA)>`

Exemple: `<!ELEMENT de (#PCDATA)>`

- **Éléments de contenus quelconques**

Les éléments déclarés avec le mot-clé de ANY peuvent contenir toute combinaison de données analysables:

`<!ELEMENT nom-element ANY>`

Exemple: `<!ELEMENT note ANY>`



Déclaration d'un élément

- **Éléments avec une séquence d'enfants**

Les éléments avec un ou plusieurs enfants sont déclarés avec le nom des éléments enfants entre parenthèses:

`<!ELEMENT nom-element (child1)>` ou `<!ELEMENT nom-elt (child1,child2,...)>`

Exemple: `<!ELEMENT note (a,de,titre,corps)>`

Lorsque les enfants sont déclarés dans une séquence séparée par des virgules, les enfants doivent apparaître dans la même séquence dans le document XML. Les enfants doivent aussi être déclarés, et peuvent avoir aussi des enfants. La déclaration complète de l'élément "note" est:

```
<!ELEMENT note (a,de,titre,corps)>
```

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

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

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

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




Déclaration d'un élément

- **Déclaration d'une seule occurrence d'un élément**

`<!ELEMENT nom-element (nom-enfant)>`

Exemple: `<!ELEMENT note (message)>`

L'exemple ci-dessus déclare que l'élément enfant «message» doit se produire une et une seule fois dans l'élément «note»..

- **Déclaration d'une occurrence minimale d'un élément**

`<!ELEMENT nom-element (nom-enfant+)>`

exemple: `<!ELEMENT note (message+)>`

Le signe + dans l'exemple ci-dessus déclare que l'élément enfant "message" doit se produire une ou plusieurs fois dans l'élément "note".



Déclaration d'un élément

- **Déclaration de zéro ou plusieurs occurrences d'un élément**

`<!ELEMENT nom-element (nom-enfant *)>`

Exemple: `<!ELEMENT note (message*)>`

Le signe * dans l'exemple ci-dessus déclare que l'élément enfant "message" peut se produire zéro ou plusieurs fois dans l'élément "note".

- **Déclaration de zéro ou une occurrence d'un élément**

`<!ELEMENT nom-element (nom-enfant?)>`

exemple : `<!ELEMENT note (message?)>`

Le signe ? déclare que l'élément enfant «message» peut se produire zéro ou une seule fois dans l'élément «note».



Déclaration d'un élément

- **Déclaration d'un contenu conditionnel (soit/ou)**

```
<!ELEMENT note (a,de,titre,(message|corps))>
```

L'exemple ci-dessus déclare que l'élément "note" doit contenir un élément "a", un élément "de", un élément "titre" et un élément "message" ou "corps".

- **Déclaration d'un contenu mixte**

```
<!ELEMENT note (#PCDATA|a|de|titre|message)*>
```

L'exemple ci-dessus déclare que l'élément "note" peut contenir zéro ou plus d'occurrences de données de caractères parsés, "a", "de", "titre" ou "message".



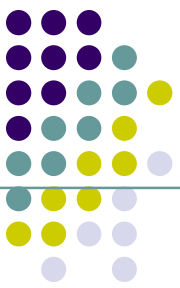
Déclaration d'un attribut

Dans une DTD, les attributs sont déclarés avec la déclaration ATTLIST.

`<!ATTLIST nom-element nom-attribut type-attribut valeur-attribut>`

Exemple DTD: `<!ATTLIST payment type CDATA "check">`

Exemple XML : `<payment type="check" />`



Déclaration d'un attribut

Le **type de l'attribut** peut être l'un des types suivants:

Type	Description
CDATA	Données caractères
(<i>en1 en2 ..</i>)	La valeur doit correspondre à une liste énumérée
ID	La valeur est un identifiant unique
IDREF	La valeur est l'id d'un autre élément
IDREFS	La valeur est une liste d'autres identifiants
NMTOKEN	La valeur est un nom XML valide
NMTOKENS	La valeur est une liste de noms XML valides
ENTITY	La valeur est une entité
ENTITIES	La valeur est une liste d'entités
NOTATION	La valeur est le nom d'une notation
xml:	La valeur est une valeur xml prédéfinie



Déclaration d'un attribut

La **valeur de l'attribut** peut être l'une des valeurs suivantes:

Valeur	Description
<i>value</i>	La valeur par défaut de l'attribut
#REQUIRED	L'attribut est requis
#IMPLIED	L'attribut est facultatif
#FIXED <i>valeur</i>	La valeur de l'attribut est fixe



Déclaration d'un attribut

- **Valeur d'attribut par défaut:**

```
<!ELEMENT carre EMPTY>  
<!ATTLIST carre largeur CDATA "0">
```

XML valide:

```
<carre largeur="100" />
```

Dans l'exemple ci-dessus, l'élément "carre" est défini comme un élément vide avec un attribut "largeur" de type CDATA. Si aucune largeur n'est spécifiée, elle a une valeur par défaut de 0.



Déclaration d'un attribut

- Champs obligatoires (**#REQUIRED**)

Syntaxe

<!ATTLIST nom-element nom-attribut type-attribut #REQUIRED>

Exemple DTD:

<!ATTLIST personne numero CDATA #REQUIRED>

XML valide : **<personne numero="5677" />**

XML invalide : **<personne />**

On utilise le mot-clé **#REQUIRED** si on n'a pas une option pour une valeur par défaut et qu'une valeur soit obligatoire.



Déclaration d'un attribut

- Champs facultatifs (#IMPLIED)

Syntaxe :

```
<!ATTLIST nom-element nom-attribut type-attribut #IMPLIED>
```

Exemple DTD:

```
<!ATTLIST contact fax CDATA #IMPLIED>
```

XML valide :

```
<contact fax="555-667788" />
```

XML valid :

```
<contact />
```

On utilise le mot-clé #IMPLIED s'il n'y a pas de valeur par défaut et qu'on ne souhaite pas imposer de valeur.



Déclaration d'un attribut

- Champs fixes (#FIXED)

Syntaxe :

```
<!ATTLIST nom-element nom-attribut type-attribut #FIXED "valeur">
```

Exemple DTD:

```
<!ATTLIST expéditeur compagnie CDATA #FIXED "Microsoft">
```

Valid XML: `<expéditeur compagnie="Gamtel" />`

Invalid XML: `<expéditeur compagnie="Africell" />`

On utilise le mot-clé #FIXED lorsqu'on souhaite attribuer une valeur fixe à un attribut sans permettre à l'auteur de le modifier. L'analyseur XML renverra une erreur si une autre valeur est mise.



Déclaration d'un attribut

Valeurs d'attribut énumérées

Syntaxe

```
<!ATTLIST nom-element nom-attribut (val1|val2|..) valeur-par-defaut>
```

Exemple

DTD:

```
<!ATTLIST payment type (check|cash) "cash">
```

Exemple XML:

```
<payment type="check" />
```

ou

```
<payment type="cash" />
```

On utilise les valeurs d'attribut énumérées lorsqu'on souhaite que la valeur de l'attribut appartienne à une liste.



Déclaration d'une entité

Les entités servent à définir des raccourcis vers des caractères spéciaux.
Les entités peuvent être déclarées internes ou externes.

- **Déclaration d'entité interne**

Syntaxe : `<!ENTITY nom-entite "valeur-entite">`

Exemple

DTD:

```
<!ENTITY writer "Nelson Mandela.">
```

```
<!ENTITY copyright "Copyright UTG.">
```

XML:

```
<author>&writer;&copyright;</author>
```

Remarque: Une entité comporte trois parties: une esperluette (&), un nom d'entité et un point-virgule (;).



Déclaration d'une entité

- Déclaration d'entité externe

Syntaxe : `<!ENTITY nom-entite SYSTEM "URI/URL">`

Exemple

DTD :

```
<!ENTITY writer SYSTEM  
"http://www.utg.gm/entities.dtd">  
<!ENTITY copyright SYSTEM  
"http://www.ugb.sn/entities.dtd">
```

XML :

```
<author>&writer;&copyright;</author>
```



Document Type Definition (DTD)

FIN