

Définition eXtended Markup Language

Un document XML est composé de **balises**, elles mêmes possédant des **attributs**. Il peut également contenir des **entités**, des **règles d'analyse**, des déclarations d'**espace de nom** (namespace), des **commentaires** et du **texte**

Prologue

```
<?xml version = "1.0" encoding="UTF-8"
standalone="yes" ?>
```

Impérativement la première ligne du fichier

version (de XML): 1.0 ou 1.1

encoding : Jeu de caractères du bloc XML (UTF8, ISO8859-1,...)

standalone : yes si aucun autre fichier n'est nécessaire

Règles de nommage balise et attribut

Un nom contient des lettres, des chiffres ou des caractères spéciaux.

Un nom ne débute pas par un nombre ou un caractère spécial.

Un nom ne commence pas par XML ou xml.

Un nom ne contient pas d'espaces.

Eviter - , ; . < > qui peuvent être mal interprétés.

Well formed (Document bien formé)

Dans la version 1.1 du XML, le prologue est bien renseigné.

Une seule balise racine.

Nom des balises et des attributs est conforme aux règles.

Balises en paires sont correctement fermées.

Valeurs des attributs sont entre guillemets simples ou doubles.

Les balises du document XML ne se chevauchent pas.

Contenu

Commentaire `<!-- commentaire -->`

Balise simple `<exemple/>`

Balise par paire `<exemple>contenu</exemple>`

Balise avec attribut `<exemple attr="valeur"></exemple>`

DTD (Document Type Definition)

La DTD peut être Interne au document ou externe (.dtd).

Elle définit les règles devant être respectées dans le corps XML.

Conformité et validation

Un document **valide** est un document **bien formé conforme à une définition**.

Un document **conforme** à une définition est un document qui **respecte toutes les règles** qui lui sont imposées dans la DTD.

DTD - ELEMENT

Syntaxe `<!ELEMENT balise (contenu)>`

personne **contient 1 nom** `<!ELEMENT personne (nom)>`

nom **contient une valeur simple** `<!ELEMENT nom (#PCDATA)>`

nom **ne contient pas de valeur** `<!ELEMENT nom EMPTY>`

nom **contient ou non une valeur** `<!ELEMENT nom EMPTY>`

personne **contient 1 nom et 1 prenom** (dans cet ordre) `<!ELEMENT personne (nom,prenom)>`

personne **contient 1 nom ou un prenom** `<!ELEMENT personne (nom|prenom)>`

personne **contient 1 nom et éventuellement un prenom** `<!ELEMENT personne (nom,prenom?)>`

répertoire **contient de 0 à n personne** `<!ELEMENT repertoire (personne*)>`

répertoire **contient de 1 à n personne** `<!ELEMENT repertoire (personne+)>`



By **Pierre LAFAY**

cheatography.com/pierre-lafay/

Not published yet.

Last updated 12th June, 2018.

Page 1 of 2.

Sponsored by **Readability-Score.com**

Measure your website readability!

<https://readability-score.com>

DTD - ATTLIST (attribut)	
Syntaxe	<code><!ATTLIST balise attribut ID mode></code>
sexe est masculin ou féminin	<code><!ATTLIST personne sexe (masculin féminin)></code>
id est un identifiant unique	<code><!ATTLIST father id ID></code>
father fait référence à une autre élément	<code><!ATTLIST father IDREF></code>
id est un identifiant unique	<code><!ATTLIST father id ID></code>
civilité est une valeur simple	<code><!ATTLIST civilite CDATA></code>
civilité est obligatoire	<code><!ATTLIST civilite CDATA #REQUIRED></code>
civilité est facultatif	<code><!ATTLIST civilite CDATA #IMPLIED></code>
civilité est par défaut Mr	<code><!ATTLIST civilite CDATA "Mr"></code>

DTD - ENTITY	
Syntaxe entité générale Remplace un élément	<code><!ENTITY nom "valeur"></code>
Exemple d'utilisation d'une entité générale	<pre><!ENTITY apple "Apple"> <telephone> <marque>&apple;</marque> <modele>iPhone X</modele> </telephone></pre>
Syntaxe entité paramètre Remplace un attribut	<code><!ENTITY % nom "valeur"></code>
Exemple d'utilisation d'une entité paramètre dans la DTD	<pre><!ENTITY % civs "civ (Mr Mme)"> <!ATTLIST personne %civs;></pre>
Syntaxe entité externe analysée (équivalent à include)	<code><!ENTITY nom SYSTEM "URI"></code>
Exemple d'utilisation d'une entité externe analysée	<pre><!ENTITY apple SYSTEM "apple.xml"> <telephone> &apple; <modele>iPhone 4</modele> </telephone></pre>
Contenu apple.xml	<code><marque>Apple</marque></code>
Les entités sont des alias permettant de réutiliser des informations dans le document	



By **Pierre LAFAY**

cheatography.com/pierre-lafay/

Not published yet.

Last updated 12th June, 2018.

Page 2 of 2.

Sponsored by **Readability-Score.com**

Measure your website readability!

<https://readability-score.com>