

XML – DTD – CSS

Aide-mémoire

Bernard JACQUEMIN
SCIMEC/CREM — UHA
Bernard (pt) Jacquemin (at) uha (pt) fr

Langage XML

Notions

Le langage XML est un langage informatique strict qui permet de structurer des données – textuelles ou autres – grâce à des balises qui marquent sémantiquement ces données.

- XML : le fichier XML contient en principe les données et les balises qui structurent ces données ;
- DTD : le fichier DTD contient les règles explicitant le balisage autorisé dans le document XML liés et la structure de ce balisage ;
- XML n'est pas un langage de représentation. Il faut lui appliquer un traitement pour afficher les données contenues dans un document XML :
 - CSS : langage de création d'une feuille de style permettant d'appliquer une mise en forme au contenu des balises pour leur affichage sur un *navigateur* ;
 - XSL : langage de transformation de données XML permettant de traiter l'information structurée pour lui donner une autre forme (notamment un format de représentation).

Syntaxe

Les règles de syntaxe :

1. Les documents XML commencent par un prologue (l'encodage peut varier) :

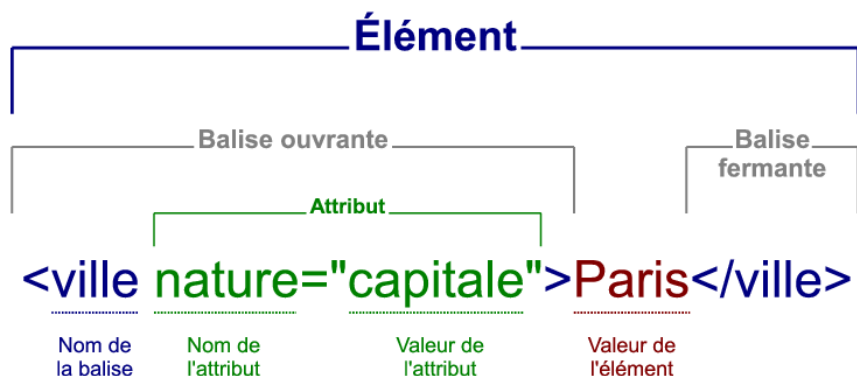
```
<?xml version="1.0" encoding="UTF-8"?>
```

2. Les documents XML ont un et un seul élément racine qui contient l'ensemble du document
3. Les noms d'éléments se présentent sous la forme d'une chaîne de caractères alphanumériques continue débutant par une lettre
4. Les éléments doivent toujours être fermés (balise fermante, balise autofermante)
5. Les éléments doivent être correctement imbriqués (pas de chevauchement)
6. Les attributs se présentent sous la forme d'un nom d'attribut (chaîne alphanumérique continue) associé à une valeur entre guillemets par le signe =
7. Seules cinq entités sont définies par défaut, et doivent être utilisées dans les documents XML à la place de leur caractère équivalent :

< ⇔ < ; " ⇔ " ;
> ⇔ > ; ' ⇔ ' ;
& ⇔ & ;

Un document conforme à ces règles est dit **bien formé**.

Vocabulaire :



La DTD : modéliser la structure des données

Une DTD (*Document Type Definition*, définition de type de document) détermine à la fois le vocabulaire et la grammaire du langage XML utilisé dans un document relié à cette DTD. Elle explicite les éléments qui peuvent apparaître dans le document XML et la nature des contenus de ces éléments (structure en arbre).

L'association entre un document XML et sa DTD s'effectue dans le document XML, avant l'élément racine, grâce à la commande suivante :

```
<!DOCTYPE eltRacine SYSTEM "cheminDTD.dtd">
```

Déclaration des éléments

Tous les éléments que contient le document XML lié doivent être déclarés et la nature de leur contenu explicitée :

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

Types de contenus :

#PCDATA : la balise concernée contient du texte (groupage autorisé, #PCDATA premier)

ANY : la balise concernée contient n'importe quel élément ou du texte (pas de groupage)

EMPTY : la balise concernée est vide (pas de groupage)

élément : la balise concernée contient un ou plusieurs éléments (groupage autorisé) : déclaration de contenu

Déclaration des contenus

Lorsqu'un élément peut (doit) contenir d'autres éléments, les modalités de contenu doivent être déclarées entre parenthèses, et éventuellement quantifiées.

Modalités de contenu :

élément simple : un seul élément requis entre parenthèses

liste d'éléments : suite ordonnée d'éléments séparés par une virgule

éléments alternatifs : éléments possibles séparés par une barre verticale (« | », ou exclusif)

Quantification :

opérateur + : 1 ou plusieurs occurrences de l'élément quantifié

opérateur ? : pas d'occurrence ou une seule occurrence de l'élément quantifié

opérateur * : élément optionnel : pas d'occurrence, une occurrence ou plusieurs occurrences permises

Le quantificateur accolé à un élément ne quantifie que cet élément. Accolé à la parenthèse fermante, il quantifie tout le contenu des parenthèses.

Déclaration des attributs

Lorsque des éléments de même nature demandent à être précisés et/ou distingués les uns des autres, ils comportent des attributs de valeurs différentes. Ces attributs demandent à être déclarés dans la DTD, et leur nature (*type*) ainsi que leurs modalités d'emploi (options) doivent être explicites :

<!ATTLIST *nomÉlément* *nomAttribut* *type* *option*>

Certaines composantes de la déclaration d'un attribut sont stables :

nomÉlément : le nom de l'élément pour lequel on déclare un attribut

nomAttribut : le nom de cet attribut

Le *typage de l'attribut* permet de qualifier la valeur qu'on peut lui donner :

CDATA : la valeur de l'attribut est du texte indéterminé

ID : la valeur de l'attribut est une chaîne de caractères qui n'apparaît qu'une seule fois comme valeur du même attribut (pour un même élément) dans l'ensemble du fichier XML

IDREF : la valeur de l'attribut correspond à un ID et fait donc référence au contenu d'un autre élément possédant un attribut de même nom et de même valeur

type énuméré : liste des valeurs autorisées pour cet attribut (entre parenthèses, séparées par une barre verticale « | »)

L'*option de l'attribut* spécifie comment l'attribut peut être utilisé :

#REQUIRED : indique que l'attribut doit obligatoirement être spécifié pour l'élément indiqué

#IMPLIED : indique que l'attribut est optionnel (utilisation permise mais pas obligatoire)

#FIXED "valeur" : indique que l'attribut doit obligatoirement apparaître avec la valeur *valeur*

"valeur" : indique que l'attribut est optionnel, et que sa valeur est *valeur* s'il n'est pas spécifié

CSS pour la présentation des données XML

La méthode la plus simple pour présenter les contenus de documents XML passe par les feuilles de style en cascade (*Cascading Style Sheet*, CSS). Il s'agit simplement d'indiquer à un navigateur quelles informations il doit afficher, et la manière de les afficher. Ces indications se font dans une feuille de style séparée du document. La feuille de style est composée de règles déterminant pour une ou plusieurs étiquettes des instructions de mise en forme.

L'association entre un document XML et sa feuille de style CSS s'effectue dans le document XML, juste après le prologue, grâce à la balise suivante :

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

Les règles CSS sont composées de

un sélecteur : fait référence à l'élément ou aux éléments pour lesquels cette règle s'applique

une ou plusieurs déclarations : couple propriété:valeur indiquant la mise en forme à appliquer

Toute déclaration doit être conclue par un point-virgule pour pouvoir s'appliquer.

```
element {
  propriete1:valeur1;
  propriete2:valeur2;
  ...
}
```

Certaines valeurs de déclarations sont récurrentes, et peuvent être appliquées à différentes propriétés. On a ainsi des valeurs de grandeurs et de couleurs :

tailles spécifiques	cm mm in	centimètre millimètre inch (pouce)	pc pt px	pica point pixel
tailles relatives	(1)em ex %	hauteur de police courante (haut du A au bas du y) hauteur de x dans l'élément courant proportion de la valeur de l'élément courant		

mots-clefs	Valeurs RGB	
● aqua	rgb(0,255,255)	rgb(0%,100%,100%)
● black	rgb(0,0,0)	rgb(0%,0%,0%)
● blue	rgb(0,0,255)	rgb(0%,0%,100%)
● fuchsia	rgb(255,0,255)	rgb(100%,0%,100%)
● gray	rgb(128,128,128)	rgb(50%,50%,50%)
● green	rgb(0,128,0)	rgb(0%,50%,0%)
● lime	rgb(0,255,0)	rgb(0%,100%,0%)
● maroon	rgb(128,0,0)	rgb(50%,0%,0%)
● navy	rgb(0,0,128)	rgb(0%,0%,50%)
● olive	rgb(128,128,0)	rgb(50%,50%,0%)
● purple	rgb(128,0,128)	rgb(50%,0%,50%)
● red	rgb(255,0,0)	rgb(100%,0%,0%)
● silver	rgb(192,192,192)	rgb(75%,75%,75%)
● teal	rgb(0,128,128)	rgb(0%,50%,50%)
white	rgb(255,255,255)	rgb(100%,100%,100%)
● yellow	rgb(255,255,0)	rgb(100%,100%,0%)

Les sélecteurs

Le sélecteur permet de désigner l'élément (ou les éléments) pour le(s)quel(s) les mises en forme décrites dans les déclarations sont demandées.

Sélecteur simple : l'élément pour lequel la/les déclaration(s) s'applique(nt)

Sélecteurs en liste : liste des éléments – *séparés par une virgule* – pour lesquels la déclaration s'applique

Sélecteur contextuel : pour que la déclaration s'applique, il faut que l'élément :

- soit le fils d'un autre élément : syntaxe de sélecteur = père > élément { ... }
- soit le descendant d'un autre élément : syntaxe de sélecteur = aïeul élément { ... }

Sélecteur en fonction de la présence d'un attribut : la déclaration s'applique si l'élément porte un attribut attr :

```
élément[attr] { ... }
```

Sélecteur en fonction de la valeur d'un attribut : la déclaration s'applique si l'élément porte un attribut de valeur val1 :

```
élément[attr="val1"] { ... }
```

Les déclarations

Affichage : propriété « display »

Les valeurs de la propriété `display` permettent de déterminer la manière dont un élément sélectionné s'affiche :

display:inline affichage de l'élément sélectionné sur la même ligne que ce qui précède et suit

display:block affichage de l'élément sélectionné sur une ligne spécifique (retour à la ligne avant et après)

display:none pas d'affichage de l'élément sélectionné

display:list-item affichage de l'élément sélectionné comme un item de liste

Affichage : récupération de l'attribut

L'attribut ne peut être affiché que avant ou après l'élément qui le porte, grâce à des sélecteurs spécifiques et une déclaration de propriété `content` :

selecteur `elt:before` – **commande** `attr(nom)` exemple `elt:before { content:"texte " attr(attribut) " texte avant élément " ; }`

selecteur `elt:after` – **commande** `attr(nom)` exemple `elt:after { content:" texte après élément " attr(attribut); }`

Affichage de police : propriété « font-family »

La propriété `font-family` indique la police ou les polices (séparées par une virgule, privilégiant la première mais se rabattant successivement sur les suivantes en cas de besoin) / la famille de polices ou les familles de polices qui seront utilisées pour présenter l'élément sélectionné.

mots-clefs génériques	noms de polices
<i>serif</i>	Times, "Times New Roman", Palatino, ...
<i>sans-serif</i>	Arial, Helvetica, Geneva, Verdana, ...
<i>cursive</i>	ZapfChancery, ...
<i>fantasy</i>	Western, Critter, ...
<i>monospace</i>	Courier, System, ...

Affichage de police : propriété « font-size »

La propriété `font-size` indique la taille (absolue ou relative) de police utilisée, soit en point, soit selon les indications ci-dessous :

Valeurs relatives à la taille de police du navigateur :

<i>xx-small</i>	la taille la plus petite possible
<i>x-small</i>	1.5 fois la taille de <i>xx-small</i>
<i>small</i>	1.5 fois la taille de <i>x-small</i>
<i>medium</i>	1.5 fois la taille de <i>small</i>
<i>large</i>	1.5 fois la taille de <i>medium</i>
<i>x-large</i>	1.5 fois la taille de <i>large</i>
<i>xx-large</i>	1.5 fois la taille de <i>x-large</i>

Valeurs relatives à la taille de police de l'élément parent :

<i>smaller</i>	33% plus petit que la police de l'élément parent
<i>larger</i>	50% plus grand que la police de l'élément parent

Affichage de police : propriété « font-style »

La propriété `font-style` indique le style d'affichage de la police. Elle peut prendre 3 valeurs différentes :

normal police droite

italic lettres penchées avec serif

oblique lettres penchées sans serif

Affichage de police : propriété « font-weight »

La propriété `font-weight` indique le gras avec laquelle la police est affichée. Elle peut prendre plusieurs valeurs différentes :

Valeurs relatives à la police de l'élément parent :

bolder plus gras que la police de l'élément parent
lighter moins gras que la police de l'élément parent

Valeurs spécifiques :

normal niveau de gras normal de la police du navigateur
bold texte en caractères gras typiques
100, 200 ... 800, 900 valeurs de niveau de gras de peu gras à très gras

Affichage de police : propriété « text-decoration »

La propriété `text-decoration` indique le placement d'une ligne horizontale liée aux caractères. Elle peut prendre 3 valeurs différentes :

`underline` : soulignement classique

`overline` : barre horizontale au dessus des caractères de l'élément concerné

`line-through` : caractères de l'élément sélectionné barrés

Affichage de police : propriété « font-variant »

La propriété `font-variant` donne la possibilité de mettre en petites capitales le contenu de l'élément sélectionné. Elle peut prendre 2 valeurs différentes :

normal : les caractères se présentent dans leur casse originale

`small-caps` : les caractères de l'élément sélectionné s'affichent en petites capitales

Affichage de police : propriété « text-transform »

La propriété `text-transform` permet de transformer tout ou partie du contenu de l'élément sélectionné en capitales. Elle peut prendre 3 valeurs différentes :

`uppercase` : toutes les lettres de l'élément sélectionné sont affichées en capitales

`lowercase` : toutes les lettres de l'élément sélectionné sont affichées en minuscules

`capitalize` : la première lettre de chaque mot de l'élément sélectionné est affichée en capitale

Affichage de police : propriété « color »

La propriété `color` donne la possibilité d'afficher les caractères de l'élément sélectionné dans une des 16 couleurs standard présentées ci-dessus.

Affichage de paragraphe : propriété « text-align »

La propriété `text-align` donne la possibilité d'aligner les contenus d'éléments sélectionnés sur les marges (bords du navigateur). Elle peut prendre 4 valeurs différentes :

left : alignement à gauche
right : alignement à droite
justify : alignement à gauche et à droite
center : centré

Affichage de paragraphe : propriété « text-indent »

La propriété `text-indent` donne la possibilité de marquer un retrait de la première ligne de l'élément sélectionné. Elle peut prendre des valeurs de dimensions absolues ou relatives (voir tableaux ci-dessus).

Affichage de paragraphe : propriété « line-height »

La propriété `line-height` permet de modifier la taille de l'interligne de l'élément sélectionné. Elle peut prendre des valeurs de dimensions absolues ou relatives (voir tableaux ci-dessus). L'interligne normal est de 1em (100%).

Affichage de page : propriété « margin »

La propriété `margin` permet de modifier la taille des marges pour l'élément sélectionné, c'est-à-dire l'espace entre le bord du navigateur et l'espace d'affichage de l'élément. Elle peut prendre des valeurs de dimensions absolues ou relatives (voir tableaux ci-dessus). Elle se décline en 5 sous-propriétés permettant de gérer l'ensemble des marges en commun, ou de les traiter séparément :

`margin` : permet de donner à toutes les marges une même valeur
`margin-left` : gère la marge de gauche
`margin-right` : gère la marge de droite
`margin-top` : gère la marge du haut
`margin-bottom` : gère la marge du bas

Affichage de page : propriété « border »

La propriété `border` permet de dessiner un cadre autour de l'élément sélectionné. Elle se décline en 3 sous-propriétés permettant de gérer l'apparence de la bordure :

`border-style` : indique le style du trait qui marque la bordure, selon 9 valeurs :

<i>solid</i>	trait continu	<i>dotted</i>	pointillés	<i>dashed</i>	tirets
<i>double</i>	trait double	<i>groove</i>	cannelure 3D	<i>ridge</i>	strie 3D
<i>inset</i>	incrustation	<i>outset</i>	extrusion	<i>none</i>	aucune

`border-width` : indique l'épaisseur du trait (absolue ou relative, voir tableau ci-dessus)

`border-color` : indique la couleur du trait (parmi les 16 couleurs prédéfinies, voir ci-dessus)

Affichage de page : propriété « background-color »

La propriété `background-color` permet de donner une couleur de fond à l'affichage de l'élément sélectionné. La couleur est à choisir notamment parmi les 16 couleurs prédéfinies (voir tableau ci-dessus).