

CSI 3540

Structures, techniques et normes du Web

La grammaire DTD de XHTML

Objectif:

- Être en mesure de lire et de comprendre la grammaire DTD de XHTML

Lectures:

- Web Technologies (2007) pages 98–106

Plan

1. Concepts de base du langage XML
2. Déclaration d'un élément
3. Déclaration d'une liste d'attributs
4. Déclaration d'une entité
5. Fichiers DTD
6. Exemple du cellier

Cette semaine

- **Youtube** (Google), puis **Vimeo** débutent l'utilisation d'**HTML5** (en particulier la balise **VIDEO** et son API)
- **Firefox** lance sa version **3.6** du populaire navigateur, apportant un support pour... **HTML5**

Motd

- Examen partiel repoussé au 23 février
(en classe)

XHTML 1.1 (Strict)

```
<?xml version="1.1" encoding="utf-8"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">  
  <head>  
    <title>CSI3540 : XHTML et son DTD</title>  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  </head>  
  <body>  
    <p>Contenu d'un élément de type P</p>  
  </body>  
</html>
```


XHTML 1.0

- 3 “saveurs” de DTD :

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```


Concepts

- **XHTML** est un langage de balisage largement utilisé pour la conception de pages Web
- **XML**, et plus particulièrement une grammaire **DTD** (définition de type de document), décrit la syntaxe du langage XHTML
- Pourquoi un métalangage ?

Concepts

- La **validation** d'un document détermine s'il est **bien formé**, si l'ordre et les contraintes d'imbrication des éléments respectent les règles spécifiques au vocabulaire
- Formalismes : **DTD** et **XML Schema**
(notation alternative : Relax NG)
- Des outils : Éditeurs tels que **<oXygen/>**
ou encore des analyseurs syntaxiques
Xerces

XML

Documents bien formés

- Un document se compose de **données textuelles** et de **balisage**
- Balisage : **balises** + **appels**
- Balises : débutent par «<» et se terminent par «>»

Documents bien formés

- **3 types de balises** : ouverture, fermeture et élément vide (ne peut contenir des données)
- Les balises d'ouverture et de fermeture doivent être **parfaitement appariées** (cet enchevêtrement n'est pas valide : `<a>`)
- En conséquent, il y a toujours **une seule balise à la racine de l'arbre** d'analyse

Documents bien formés

- **Appels** :
«&» + appel de caractère + «;» ou
«&» + appel d'entité + «;»
- Appels d'entité **prédéfinis** : < (<), > (>), & (&), ' (') et " (")
- Un document **XML** contenant des données s'appelle **instance**

Survol d'un DTD

-//W3C//DTD XHTML 1.0 Strict//EN

<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>

<!-- Le DTD contient 3 types d'informations : ELEMENT, ATTLIST et ENTITY (ainsi que des commentaires) -->

<!ELEMENT html (head, body)>

<!ATTLIST html

lang NMTOKEN #IMPLIED

xml:lang NMTOKEN #IMPLIED

dir (ltr|rtl) #IMPLIED

id ID #IMPLIED

xmlns CDATA #FIXED '<http://www.w3.org/1999/xhtml>'>

<!ENTITY copy "©">

<!ELEMENT ...>

- <!ELEMENT html (head, body)>
- Chaque élément du langage définit par le DTD est introduit à l'aide d'une «déclaration de type d'élément»
- À la suite du nom de l'élément, ici HTML, il y a la «spécification du contenu». La déclaration ci-haut indique que l'élément HTML a deux fils, HEAD et BODY.

<!ATTLIST ...>

```
<!ATTLIST html
  lang      NMTOKEN #IMPLIED
  xml:lang  NMTOKEN #IMPLIED
  dir       (ltr|rtl) #IMPLIED
  id        ID       #IMPLIED
  xmlns     CDATA    #FIXED 'http://www.w3.org/1999/xhtml'>
```

- «Déclaration d'une liste d'attributs»
- L'élément HTML possède 4 attributs :
LANG, XML:LANG, DIR, ID et XMLNS

<!ENTITY ...>

<!ENTITY copy "©">

- «Déclaration d'entité»
- Les déclarations d'entité sont en quelque sorte des macros commandes
 - La déclaration associe un nom à une chaîne de caractères
 - Les occurrences de `©` sont remplacées par la chaîne `©` (les macro-expansions se poursuivent jusqu'à évaluation complète du document)
- «Entité générale»

<!ENTITY ...>

<!ENTITY % URI "CDATA">

- «Entité paramètre» préfixé par %
- On y réfère qu'à l'intérieur du DTD, en utilisant le préfixe %
 - %URI; est remplacé par CDATA

DTD, Entités et XHTML

- Dans le but de faciliter la compréhension du DTD de XHTML, **plus de 50 entités ont été définies**
- À première vue, ça peut sembler étrange, mais plusieurs noms sont associés à la même valeur (voir page suivante)
- Cet exercice n'apporte rien aux analyseurs syntaxiques


```
<!ENTITY % ContentType "CDATA">  
  <!-- media type, as per [RFC2045] -->
```

```
<!ENTITY % ContentTypes "CDATA">  
  <!-- comma-separated list of media types, as per [RFC2045] -->
```

```
<!ENTITY % Charset "CDATA">  
  <!-- a character encoding, as per [RFC2045] -->
```

```
<!ENTITY % Charsets "CDATA">  
  <!-- a space separated list of character encodings, as per [RFC2045] -->
```


Présentation détaillée

DTD XML et XHTML 1.0 en particulier

DTD

- La **validation** d'un document XML (d'une instance) a pour but de s'assurer que les règles d'**imbrication** et l'**ordre** des éléments sont conformes à la spécification
- Le **DTD** est l'un des formalisme pour décrire cette **spécification**
- Malheureusement, le **DTD** n'est pas lui-même un document XML

DTD

- Les déclarations de type d'élément
- Les déclarations de liste d'attributs
- Les appels de caractère et d'entité

Les déclarations de type d'élément

- **<!ELEMENT Nom SpécContenu >**
- Le nom de l'élément : **a, p, table, br**, etc.
- Une spécification de contenu

Contenu d'éléments

ANY n'est pas utilisé pour la spécification de XHTML

Type	Syntaxe
Données textuelles	#PCDATA
Sans contenu	EMPTY
Séquence ordonnée	(elt1, elt2, ...)
Choix	(elt1 elt2 ...)
Contenu mixte	(#PCDATA elt1 elt2 ...)
Sans restriction	ANY

PCDATA

- **PCDATA** signifie Parsed Character DATA
- Chaîne de caractères pouvant contenir des appels d'entités
- Le processeur doit faire l'expansion des références d'entités (parsing)
- **CDATA** signifie Character Data ; sans traitement

Exemples d'éléments sans contenu

- `<!ELEMENT img EMPTY>`

`<!ATTLIST img`

`src %URI; #REQUIRED`

`alt %Text; #REQUIRED`

`longdesc %URI; #IMPLIED`

`height %Length; #IMPLIED`

`width %Length; #IMPLIED`

`...>`

XML

- ``

XHTML

Itérateurs

Semblables
aux expressions
régulières

Symbole	Description
?	Zéro ou une occurrences
*	Zéro ou plusieurs occurrences
+	Une ou plusieurs occurrences

Itérateur : ?

- `<!ELEMENT table
(caption?, (col* | colgroup*), thead?, tfoot?, (tbody+ | tr+))>`

```
<table border="1">  
  <caption>  
    Représentation RGB des Couleurs  
  </caption>  
  <tr>  
    <th>Rouge (R)</th><th>Vert (G)</th><th>Bleu (B)</th><th>Couleur</th>  
  </tr>  
  ...  
</table>
```


Itérateur : +

- `<!ELEMENT ul (li)+>`
- ``
 - `Premier item`
 - `Deuxième item```

Itérateur : +

```
<?xml version="1.1" encoding="utf-8"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">  
  <head>  
    <title>CSI3540: DTD</title>  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  </head>  
  <body>  
    <ul>  
      <li>Une liste ayant un élément.</li>  
    </ul>  
    <ul>  
    </ul>  
  </body>  
</html>
```




Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Notes and Potential Issues](#) [Validation Output](#)

Error found while checking this document as XHTML 1.1!

Result: 1 Error, 1 warning(s)

Source:

```
<title>CSI3540: DTD</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>
  <ul>
    <li>Une liste ayant un élément.</li>
  </ul>
  <ul>
  </ul>
</body>
</html>
```

Encoding: utf-8

Doctype: XHTML 1.1

Root Element: html

Root Namespace: <http://www.w3.org/1999/xhtml>



The W3C validators rely on community support for hosting and development.

[Donate](#) and help us build better tools for a better web.

Itérateur : +

- Line 16, Column 9: end tag for "ul" which is not finished
- ``
- Most likely, you nested tags and closed them in the wrong order. For example `<p>...</p>` is not acceptable, as `` must be closed before `<p>`. Acceptable nesting is: `<p>...</p>`
- Another possibility is that you used an element which requires a child element that you did not include. Hence the parent element is "not finished", not complete. For instance, in HTML the `<head>` element must contain a `<title>` child element, lists (`ul`, `ol`, `dl`) require list items (`li`, or `dt`, `dd`), and so on.

nonvalid.html [/Users/turcotte/Sync/Teaching/CSI3540/2010/Sandbox/Src/04_DTD/01_students/nonvalid.html] - <oxygen/> XML Editor

External Tools Saxon-EE

XPath 2.0

Project

- sample.xpr
 - css
 - debugger
 - fo
 - import
 - jsp
 - nvd1
 - relaxng
 - schematron
 - svg
 - wsdl
 - xquery

Outline

- html "http://www.w3.org/1999/xhtml"
 - head CSI3540: DTD
 - body

```
1 <?xml version="1.1" encoding="utf-8"?>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
4   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
5
6 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
7   <head>
8     <title>CSI3540: DTD</title>
9     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
10  </head>
11  <body>
12    <ul>
13      <li>Une liste ayant un élément.</li>
14    </ul>
15    <ul>
16    </ul>
17  </body>
18 </html>
19
```

E [Xerces] The content of element type "ul" is incomplete, it must match "(li)+".

Text Grid Author

/Users/turcotte/Sync/Teaching/CSI3540/2010/Sandbox/Src/04_D... U+003C 1:1

Déclarations de contenu imbriquées

- `<!ELEMENT table (caption?, (col* | colgroup*), thead?, tfoot?, (tbody+ | tr+))>`

```
<table border="1">
  <caption>
    Représentation RGB des Couleurs
  </caption>
  <tr>
    <th>Rouge (R)</th><th>Vert (G)</th><th>Bleu (B)</th><th>Couleur</th>
  </tr>
  ...
</table>
```


Itérateurs

- Les déclarations qui suivent sont-elles équivalentes?
 - `<!ELEMENT table (caption?, (col* | colgroup*), thead?, tfoot?, (tbody+ | tr+))>`
 - `<!ELEMENT table (caption?, (col | colgroup)*, thead?, tfoot?, (tbody+ | tr+))>`

<!ATTRLIST ...>

- <!ELEMENT img EMPTY>

<!ATTRLIST img

| | | |
|----------|----------|-----------|
| src | %URI; | #REQUIRED |
| alt | %Text; | #REQUIRED |
| longdesc | %URI; | #IMPLIED |
| height | %Length; | #IMPLIED |
| width | %Length; | #IMPLIED> |

Nom

Valeur par défaut

Type de données

Appels d'entité

- `<!ELEMENT img EMPTY>`
`<!ATTLIST img`

<code>src</code>	<code>%URI;</code>	<code>#REQUIRED</code>
<code>alt</code>	<code>%Text;</code>	<code>#REQUIRED</code>
<code>longdesc</code>	<code>%URI;</code>	<code>#IMPLIED</code>
<code>height</code>	<code>%Length;</code>	<code>#IMPLIED</code>
<code>width</code>	<code>%Length;</code>	<code>#IMPLIED</code>
<code>...></code>		

Appels d'entité

Après
remplacement des
entités

- `<!ELEMENT img EMPTY`
`<!ATTLIST img`

<code>src</code>	<code>CDATA</code>	<code>#REQUIRED</code>
<code>alt</code>	<code>CDATA</code>	<code>#REQUIRED</code>
<code>longdesc</code>	<code>CDATA</code>	<code>#IMPLIED</code>
<code>height</code>	<code>CDATA</code>	<code>#IMPLIED</code>
<code>width</code>	<code>CDATA</code>	<code>#IMPLIED</code>
<code>... ></code>		

Type d'attributs

Syntaxe	Description
CDATA	Données textuelles (sauf < et &)
ID	Identificateur unique
IDREF	Référence à un id. unique
NMTOKEN	Nom (identifiant) (lettres, chiffres, ".", "-", "_" et ":")

Attribut id

- `<p id="DSCN3371">`
 `<p id="DSCN3371">`
 Ceci n'est pas un document XML valide, pourquoi?
 `</p>`
`</p>`

Type d'attributs (suite)

Syntaxe	Description
(val1 val2 ...)	Liste d'attributs possibles
IDREFS	Liste de Références à des attributs id uniques (séparés par des espaces)

Attributs communs

<!ENTITY % attrs

| | | |
|-------|--------------|-----------|
| id | ID | #IMPLIED |
| class | CDATA | #IMPLIED |
| style | %StyleSheet; | #IMPLIED |
| title | %Text; | #IMPLIED" |

>

<!ELEMENT link EMPTY>

<!ATTLIST link

%attrs;

| | | |
|----------|----------------|----------|
| charset | %Charset; | #IMPLIED |
| href | %URI; | #IMPLIED |
| hreflang | %LanguageCode; | #IMPLIED |
| type | %ContentType; | #IMPLIED |
| rel | %LinkTypes; | #IMPLIED |
| rev | %LinkTypes; | #IMPLIED |
| media | %MediaDesc; | #IMPLIED |

>

Attributs ID et IDREF

```
<form action=".." method="post">
<table>
  <tr>
    <td><label for="label_prenom">Pr&eacute;nom</label></td>
    <td><input type="text" name="prenom" id="label_prenom"></td>
  </tr>
</table>
</form>
```

```
<!ELEMENT label %Inline;>
```

```
<!ATTLIST label
```

```
...
```

```
for          IDREF          #IMPLIED
```

```
...
```

```
>
```


Valeur par défaut

Syntaxe	Description
#REQUIRED	Aucune valeur par défaut, cet attribut est obligatoire
#IMPLIED	Aucune valeur par défaut, cet attribut est optionnel
#FIXED	Valeur prédéterminée
Litéral	Spécifie la valeur par défaut

Valeur par défaut

- On peut aussi inclure une valeur par défaut

```
<!ATTLIST form
```

```
...  
method (get|post) "get"  
...>
```


Valeur obligatoire

- `<!ELEMENT img EMPTY>`

`<!ATTLIST img`

`src CDATA #REQUIRED`

`alt CDATA #REQUIRED`

`longdesc CDATA #IMPLIED`

`height CDATA #IMPLIED`

`width CDATA #IMPLIED`

`... >`

- ``

Valeur par défaut

- `<!ATTLIST html`
 `%i1 8n;`
 `id ID #IMPLIED`
 `xmlns %URI; #FIXED 'http://www.w3.org/1999/xhtml'`
 `>`

NMTOKEN

- `Go to introduction`
...
`<h1>Introduction</h1>`

Entités externes

- XML possède un mécanisme afin d'inclure des déclarations d'entités sauvegardées dans un fichier externe
- ```
<!ENTITY % HTMLlat1 PUBLIC
"-//W3C//ENTITIES Latin 1 for XHTML//EN"
"xhtml-lat1.ent">
%HTMLlat1;
```
- La dernière ligne force l'incusion du fichier xhtml-lat1.ent dans le fichier courant (ici xhtml1-strict.dtd)



```
<!ENTITY % HTMLlat1 PUBLIC
 "-//W3C//ENTITIES Latin 1 for XHTML"
 "xhtml-lat1.ent">
%HTMLlat1;
```

lat1 pour les caractères  
accentués,

special pour quot, amp, lt, etc.

symbol pour les lettres  
grecques, flèches, etc.

```
<!ENTITY % HTMLsymbol PUBLIC
 "-//W3C//ENTITIES Symbols for XHTML//EN"
 "xhtml-symbol.ent">
%HTMLsymbol;
```

```
<!ENTITY % HTMLspecial PUBLIC
 "-//W3C//ENTITIES Special for XHTML//EN"
 "xhtml-special.ent">
%HTMLspecial;
```



# Remarques

- Le DTD peut se trouver dans le fichier XML ou dans un fichier séparé
- Le DTD dans un fichier séparé peut être partagé par plusieurs instances, et ainsi réduire la taille des instances
- Généralement un fichier dont l'extension est .dtd



# Remarques

- La déclaration de type du fichier XHTML comprend 3 choses : 1) l'élément racine de l'instance, 2) un «identificateur publique formel» (en format URN) et 3) un «identificateur système» :

- `<!DOCTYPE html`

```
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```



# Remarques

- En raison des nombreux outils développés,
- de la vaste base d'utilisateurs
- et des assises solides du langage,
- XML est maintenant un standard très répandu en dehors des applications Web



# Exemple complet

- Cellar.dtd du document de Guy Lapalme



```
<cellar>
 <wine code="C00043125">
 <purchaseDate>2005-06-20</purchaseDate>
 <quantity>2</quantity>
 <comment>
 <bold>&G</bold>: should reorder soon
 </comment>
 </wine>
 ...
 <wine code="C00929026">
 <purchaseDate>2003-10-15</purchaseDate>
 <quantity>1</quantity>
 <comment>for <bold>big</bold> parties</comment>
 </wine>
</cellar>
```



<!ELEMENT cellar (wine)\* >

<!ELEMENT wine (purchaseDate,quantity,rating?,comment?) >

<!ATTLIST wine code IDREF #REQUIRED >

<!ELEMENT purchaseDate (#PCDATA) >

<!ELEMENT quantity (#PCDATA) >

<!ELEMENT rating EMPTY >

<!ATTLIST rating stars CDATA #IMPLIED>

<!ENTITY % Comment "(#PCDATA | emph | bold)\*" >

<!ELEMENT emph (#PCDATA) >

<!ELEMENT bold (#PCDATA) >

<!ENTITY GL "Guy Lapalme" >



# Revue des concepts

- Le DTD comprend 3 types d'informations : ELEMENT, ATTLIST et ENTITY
- Les déclarations d'entités ressemblent à des macros commandes
- Vous devriez être en mesure de lire et comprendre le DTD de XHTML 1.0



# Pour le prochain cours

- Pour le prochain cours, vous devriez lire le chapitre 3 (pages 121 à 177) qui porte sur les feuilles de style en cascade



# Ressources

- Le langage de balisage extensible (XML) 1.1  
[ <http://www.yoyodesign.org/doc/w3c/xml11/> ]  
Visité à l'été 2007
- Extensible HTML version 1.0 Strict DTD  
[ <http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd> ] Visité à l'été 2007
- XML: Looking at the Forest Instead of the Trees  
[ <http://www.iro.umontreal.ca/~lapalme/ForestInsteadOfTheTrees> ] Visité à l'été 2007