

INU3011 Documents structurés

Cours 4

Validité en XML :
mécanique, bénéfiques, expérience-
autrice, écosystème autour d'un modèle

Plan

- Retour sur Mini-quiz 1
- Validité en XML
 - Retour sur la lecture "Validité"
- StudiUM
 - TP 2 - Travail individuel
 - Encodage d'un texte conformément à un modèle XML donné
 - Exercices après lecture "Langage des DTD"

Mini-quiz 1

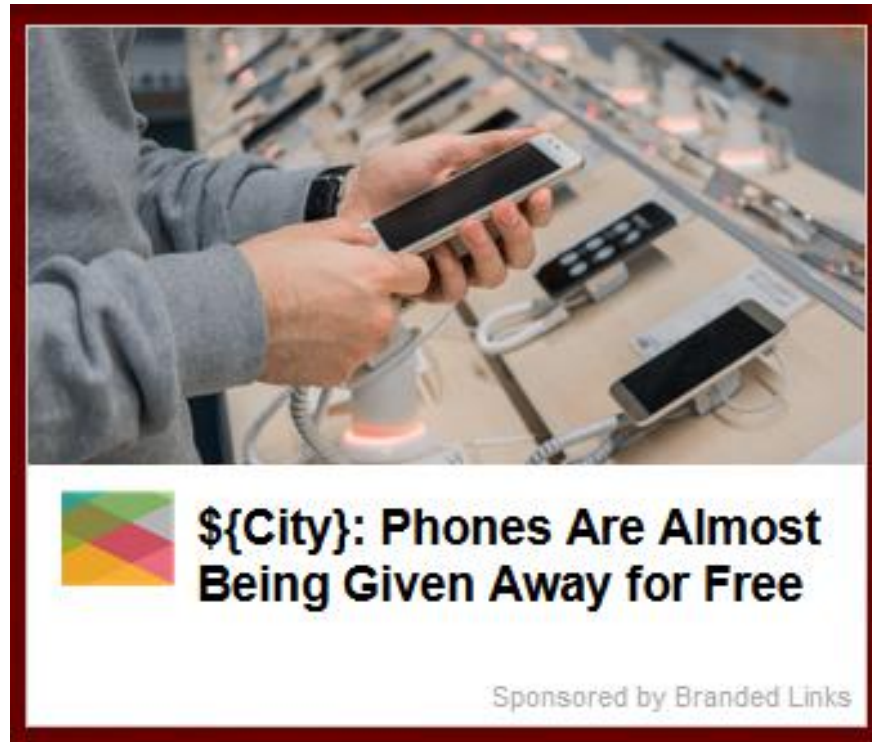
- Moyenne $42,5 / 50 = 85\%$
- Q1 : nombre de caractères
- Q6 : SVG
- Q10 : Un commentaire ne peut jamais contenir un élément ou une balise
- Q16 : Élément conteneur

- Questions ?

Validité en XML

Vue d'ensemble
et bénéfiques

Bénéfice de la rigueur du bien-formé XML



Ce que permet la validité (1/3)

- La validité permet à une *architecte de documents** de formuler des règles syntaxiques *au-delà du bien-formé*, qui définissent un certain *type de documents*
- Exemples: * Ou *modélisatrice*, ou "*document engineer*"
 - article d'une revue scientifique
 - plan de cours dans une institution donnée
 - édition critique annotée d'une œuvre littéraire
 - etc.

Ce que permet la validité (2/3)

- L'intérêt de définir ainsi un type de documents est qu'il est possible de *vérifier qu'un document proposé par une autrice (ou en général une collaboratrice) a bel et bien la forme requise pour qu'on puisse lui appliquer les traitements attendus*
 - Par exemple, qu'on pourra publier correctement un article proposé à une revue scientifique

Ce que permet la validité (3/3)

- La disponibilité du modèle XML défini par la modélisatrice pour le type de documents permet de *rejeter* les documents *non conformes* avant même d'essayer de les traiter
 - Par exemple, retourner à son autrice un article non conforme soumis à une revue scientifique

Trois niveaux de règles syntaxiques applicables à un document XML valide

- Validité XML
- Bien-formé XML
- Fichier texte (Unicode-UTF-8)

Effet de la validité

- Les restrictions auxquelles les documents valides sont soumis font en sorte que leur contenu est *prévisible* (contrairement aux documents qui ne sont que bien formés)
- Le *degré* de cette prévisibilité est décidé par la modélisatrice et peut être aussi grand (ou petit) que jugé nécessaire par elle

Validité en XML

Mécanique

Mécanique de la validité (1/2)

- Les règles de validité restreignent :
 - quels *types* (noms) d'éléments on a le droit d'utiliser dans les documents
 - où dans les documents on a le *droit* ou *l'obligation* d'utiliser chaque type d'éléments
 - quels attributs on *peut* ou *doit* spécifier pour chaque type d'éléments

Mécanique de la validité (2/2)

- Ces restrictions sont décrites dans une *DTD : Document Type Definition*
- Créée par la modélisatrice des documents
 - qui peut être la gestionnaire de l'information elle-même, ou mandatée par celle-ci
- Mise en ligne (p.ex. sur internet) par la gestionnaire, à l'intention des autrices

Du côté de l'autrice

- Une *déclaration de type de document* (DOCTYPE) est ajoutée par l'autrice dans le prologue du document en création :
 - cette déclaration pointe à la DTD (URL), et
 - indique le nom d'élément que l'élément-document doit porter
- Ces deux informations sont communiquées par la modélisatrice

Validité en XML

Bénéfices

Bénéfices de la prévisibilité

- La *prévisibilité* des documents valides fait que plusieurs mécaniques (simples) de traitement *uniforme* sont applicables :
 - Stylage CSS (*même langage* que pour HTML)
 - Stylage XSLT
 - XPath
- En général : développement d'applications est de beaucoup facilité par la prévisibilité
 - Base de données XML

Validité en XML

Exemples de traitements
uniformes rendus possibles par la
validité

Exemples (1/2)

- Feuille de style CSS
 - [050-Ex-utilite-validite](#)
 - memos-CSS
 - proces-verbaux-CSS
- Feuille de style XSLT
 - [050-Ex-utilite-validite](#)
 - messages-telephoniques-XSLT

Note : lien vers feuille de style

- Un lien vers une feuille de style est ajouté dans un document XML
 - En ajoutant cette ligne *dans le prologue* :

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

- où *feuille.css* est remplacée par l'URL (relative, dans les cas simples) de la feuille à appliquer (*CSS* → *XSL* pour feuille XSLT)

*Ce genre de construction est une *instruction de traitement*, non vu dans le *Premier tour d'horizon de XML*

Exemples (2/2)

- Base de données XML
 - [Base de données vins](#)
- XPath sur fichier unique
 - Liste des candidats retenus au barreau californien

Du côté de l'autrice...

Guidance syntaxique (1/2)

- La présence de la déclaration DOCTYPE permet à un éditeur XML (comme oXygen) de prendre connaissance des règles de validité et d'aider l'autrice à créer un document valide
 - En lui proposant en tout endroit du document que les éléments ou attributs permis
 - En insérant automatiquement ce qui est obligatoire d'après la DTD

Guidance syntaxique (2/2)

- Il devient alors facile pour l'auteur de créer un document syntaxiquement valide selon la DTD
- Exemple :
 - [050-Ex-utilite-validite](#)
 - memos-CSS

Volet sémantique de la validité

- Évidemment, ce qu'on attend d'une autrice n'est pas seulement un document *valide*, mais un document **sémantiquement conforme**, c'est-à-dire...
 - où les différents éléments et attributs :
 - contiennent les informations auxquelles on s'attend (i.e., celles prévues par la modélisatrice)
 - contiennent ces informations inscrites **sous la forme attendue** (Nom, Prénom; AA-MM-JJ; etc.)

Guidance sémantique (1/5)

- La conformité sémantique d'une document *ne peut pas être vérifiée automatiquement*
- On peut cependant fournir à l'auteur des *directives* ou *indices* propres à l'amener à produire un document **sémantiquement conforme**
- On appelle **guidance sémantique** de tels indices ou directives

Guidance sémantique (2/5)

- Une première forme de guidance sémantique peut venir de la DTD elle-même :
 - Le *nom* des éléments et attributs peuvent constituer un *indice* sur ce qu'ils doivent contenir
 - Exemple : [050-Ex-utilite-validite](#)
 - memos-CSS vs messages-téléphoniques-XSLT

Guidance sémantique (3/5)

- L'*imbrication* des éléments peut donner des indices supplémentaires
 - Exemple : [050-Ex-utilite-validite](#)
 - Éléments nom dans aide-aux-autrices
- Deuxième forme de guidance sémantique
 - Un *stylage* CSS peut être appliqué la création d'un document et aider l'autrice à comprendre comment l'information saisie est destinée à être interprétée : c'est le **mode auteur**
 - Exemple : [050-Ex-utilite-validite](#) aide-aux-autrices

Guidance sémantique (4/5)

- Une troisième forme de guidance sémantique prend la forme d'*infobulles d'aide à la saisie*
 - Une *extension* de la mécanique de la validité permet à la modélisatrice d'intégrer **dans sa DTD même** des *textes d'aide à la saisie*, qui seront présentés à l'autrice sous forme d'*infobulles* pendant la création d'un document

Guidance sémantique (5/5)

- Ces infobulles apparaissent quand le curseur survole un *nom d'élément ou d'attribut*
- Elles sont un endroit idéal pour indiquer à l'autrice :
 - Des règles d'écriture particulières pour l'élément ou l'attribut
 - Des exemples de contenus attendus
- Exemple :
 - [050-Ex-utilite-validite](#) aide-aux-autrices

Infobulles d'aide à la saisie

- Concrètement, les textes des infobulles d'aide à la saisie sont inscrits par la modélisatrice directement *dans la DTD*, sous forme de commentaires XML d'une forme spéciale (commençant par "doc:")
 - Exemples :
 - [050-Ex-utilite-validite](#) aide-aux-autrices

Documentation du modèle

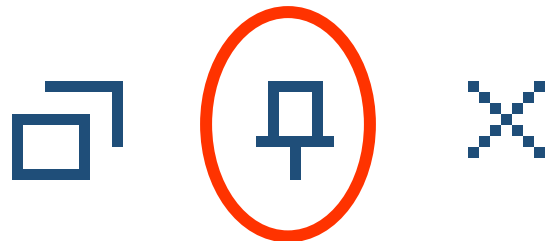
- Pour un modèle complexe, la guidance sémantique ultime vient d'une documentation séparée du modèle XML dans son ensemble, qui en décrit les objectifs, les principes de conception, etc.
 - Exemples :
 - Les "guidelines" pour la TEI
 - DocBook : The Definitive Guide
 - etc.

Réglages utiles pour les autrices dans oXygen (1/2)

- Options → Préférences
 - Éditeur → Autocomplétion
 - Insérer les attributs requis
 - Éditeur → Autocomplétion → Annotations
 - Afficher les annotations dans l'assistant d'autocomplétion
 - Afficher les annotations dans des infobulles
 - Privilégier les commentaires DTD commençant par "doc:" comme annotations

Réglages utiles pour les autrices dans oXygen (2/2)

- Fenêtre → Maximiser la zone d'édition
- Fenêtre → Afficher la vue → Éléments
- Fenêtre → Afficher la vue → Attributs
- Si désiré, fixer puis déplacer ces deux vues :



"Écosystème" d'un modèle XML normalisé

- Ressources partagées :
 - feuilles de style, documents-types, documentation (incluant textes d'infobulle), ressources pédagogiques, formation
 - autres applications de traitement spécialisées (visualisation, publication, ...)
- Réseaux d'entraide pour auteurs (encodeurs)
 - Ex.: TEI avec la [liste tei-fr](#)

Alternatives aux DTD (1/3)

- Il existe plusieurs autres langages permettant de définir des contraintes syntaxiques au-delà du bien-formé pour les documents XML
- On les appelle collectivement des "langages de contraintes XML"
- Tous ces formalismes ont comme but d'augmenter la prévisibilité des documents

Alternatives aux DTD (2/3)

- Tous ces langages intègrent des formes de guidance sémantique, dont la possibilité de fournir à l'autrice des infobulles d'aide à la saisie
- Exemples (les plus populaires) :
 - Schémas XML du W3C (recom. du W3C)
 - RelaxNG (ISO/IEC 19757-2:2003)
 - Schematron (ISO/IEC 19757-3:2020)

Alternatives aux DTD (3/3)

- Un ensemble de contraintes défini par une modélisatrice (sous forme de DTD ou dans un autre langage de contraintes) peut porter tous les synonymes suivants :
 - modèle XML
 - schéma XML
 - vocabulaire XML
 - dialecte XML
 - format XML spécialisé
 - en anglais : *tagset*, *tag library*, *tag suite*

Lien document → modèle

- Une déclaration DOCTYPE ne peut lier un document qu'à une DTD
- D'autres conventions existent pour lier un document à un modèle utilisant un autre langage de contraintes que celui des DTD
 - Exemples :
 - `<?xml-model ... ?>` dans le prologue
 - attribut `xsi:schemaLocation` sur l'élément-document
 - etc.

DTD « PUBLIC »

`PUBLIC "identifiant public" "URL"`

- S'utilise au lieu de `SYSTEM "URL"` dans la déclaration `DOCTYPE`
- Indique que la DTD pointée par l'URL est "publique" et que l'application (p.ex. oXygen) peut utiliser une copie locale de la DTD, plutôt que de la récupérer du Web
- Seulement pour certaines DTD précises
 - Par exemple XHTML, DocBook 4 Article, etc.

StudiUM