

Réseaux 1

TP 12 - Documents XML

Objectif : introduire le langage XML, créer un document XML bien formé, créer un document XML valide, afficher un document XML à l'aide de feuilles de style CSS ou XSL.

Note : vous placerez vos fichiers sur nyx, dans un répertoire ~/web/rx1/TD12

1. Créer un document XML bien formé

- 1.1. Dans un fichier de texte nommé `pizzas.xml` saisir le document XML ci-dessous. Ce document décrit plusieurs types de pizzas selon la norme PML (Pizza Markup Language)

```
<?xml version="1.0" encoding="UTF-8" ?>
<pizzamenu>
  <pizza>
    <topping extracheese="yes"> Jambon </topping>
    <price> 12,99 </price>
    <size> grande </size>
  </pizza>
  <pizza>
    <topping extracheese="no"> Chorizo </topping>
    <price> 11,99 </price>
    <size> petite </size>
  </pizza>
  <pizza>
    <topping> Champignons </topping>
    <price> 12,99 </price>
    <size> moyenne </size>
  </pizza>
  <pizza>
    <topping extracheese="yes"> Poivrons </topping>
    <price> 11,99 </price>
    <size> moyenne </size>
  </pizza>
</pizzamenu>
```

- 1.2. Visualisez votre document XML dans IE.
- 1.3. Remplacez la garniture d'une pizza au choix par "Gruyère".

2. Créer un document XML valide

- 2.1. Dans un nouveau fichier de texte nommé `pizzas.dtd`, définissez une DTD pour le document `pizzas.xml`.
- un élément `pizza` doit être composé des trois éléments dans l'ordre : garniture, prix et taille.
 - l'attribut « `extracheese` » de l'élément `pizza` ne prend que les valeurs « `yes` » ou « `no` » par défaut.
- 2.2. Intégrez l'appel à la DTD dans le prologue de `pizzas.xml`. Tester.

- 2.3. Vérifiez la validité de votre document XML à l'aide de `validateXML.html` dans `SupportCours / Réseaux1`.
Copier la page `validateXML.html` dans votre répertoire de travail, puis l'ouvrir dans IE.
Nous utilisons dans cette page le parseur XML de Microsoft.
- 2.4. Pour s'assurer que le parseur effectue la vérification, testez les cas d'erreurs suivants :
- valeur d'attribut `extracheese` erroné, différent de `yes` ou `no`
 - un élément `topping`, `price` ou `size` manquant, ou pas dans le bon ordre
- 2.5. Ajoutez dans `pizzas.dtd` un nouvel élément « header » pour contenir le nom de la pizzeria. Cet élément devra apparaître en-tête d'un document.
- 2.6. Définissez une entité « `pizzastore` » représentant le nom de la pizzeria. Tester.

3. Afficher un document XML à l'aide d'une feuille de style CSS

- 3.1. Définissez une feuille de style `pizzas.css` afin d'afficher chaque élément `pizza` dans un bloc. Chaque bloc `pizza` sera précédé d'un espace de 18 points, avec une marge à gauche et une marge à droite de 20%. Les données relatives à une `pizza` seront centrées les unes sous les autres à l'intérieur d'un bloc. Choisissez les couleurs de fond, polices de caractères ...
- 3.2. Intégrez l'appel à la feuille de style dans le prologue de `pizzas.xml`. Tester.

4. Afficher un document XML à l'aide d'une feuille de style XSL (Extensible Stylesheet Language)

- 4.1. Définissez une feuille de style `pizzas.xsl` afin de produire l'affichage sous la forme ci-dessous (nom de la pizzeria en en-tête `<h1>`, noms des éléments en gras, éléments de `pizza` séparés par des sauts de ligne, éléments `pizza` séparés par des marques de paragraphe)

Villa pizza

Garniture : Jambon
Prix : 12,99
Taille : grande

Garniture : Chorizo
Prix : 11,99
Taille : petite

Garniture : Champignons
Prix : 12,99
Taille : moyenne

Garniture : Poivrons
Prix : 11,99
Taille : moyenne

- 4.2. Dans le prologue de `pizzas.xml`, mettez en commentaire l'appel à la feuille de style CSS et intégrez l'appel à la feuille de style XSL à la place. Tester.

- 4.3. Définissez une deuxième feuille de style `pizzas2.xsl` afin de produire l’affichage sous la forme d’une table html comme ci-dessous. Tester.

Villa pizza

Garniture	Prix	Taille
Jambon	12,99	grande
Chorizo	11,99	petite
Champignons	12,99	moyenne
Poivrons	11,99	moyenne

5. Quiz XML

Testez vos connaissances sur :

http://www.w3schools.com/xml/xml_quiz.asp