

HTML (Séance 4) / XML (Séance 1)

Table des matières

I. Structure arborescente de HTML.....	1
Exercice.....	1
II. Introduction au XML.....	2
1. Différences entre XML et HTML.....	2
2. Structure d'un fichier XML.....	2
Exercice.....	3

I. Structure arborescente de HTML

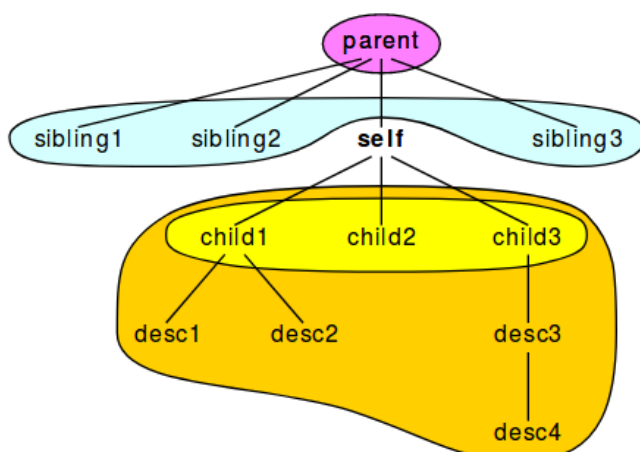
Comme les balises de HTML sont incluses les unes dans les autres, on a une structure « en arbre » du fichier HTML. On dit qu'une balise 1 est le parent de la balise 2 (ou que la balise 2 est l'enfant de la balise 1) si on a : `<balise1> ... <balise2> ... </balise2> ... </balise1>`.

On peut généraliser cette approche quasi-généalogique :

```

<parent>
  <sibling1> ... </sibling1>
  <sibling2> ... </sibling2>
  <self>
    <child1> ... <desc1></desc1> ... <desc2></desc2> ... </child1>
    <child2> ... </child2>
    <child3> ... <desc3><desc4> ... </desc4></desc3> ... </child3>
  </self>
  <sibling3> ... </sibling3>
</parent>

```



Exercice

Récupérez le fichier HTML `structure.html` sur le site du cours. Dessinez sa structure en arbre.

II. Introduction au XML

XML = **eXtensible Markup Language** (langage de balisage extensible)

Langage créé en 1996

But : échange automatisé de contenus complexes

Principes généraux :

- la structure d'un document XML est **définie et validable** par un schéma
- un document XML est **transformable** en un autre document XML

Vous connaissez déjà peut-être des applications de XML : flux RSS, format EPUB... Nous les verrons plus en détails plus tard.

1. Différences entre XML et HTML

Remarque : HTML est une « variation » de XML, un peu plus laxiste sur la syntaxe, mais plus stricte sur le vocabulaire :

HTML : les balises sont « fixes », issues d'un dictionnaire qu'on ne contrôle pas / XML : on peut créer ses propres balises (en fait, on va même créer son propre dictionnaire)

HTML : le navigateur web est « laxiste » (il ne râle pas trop si on oublie de fermer des balises, si on n'écrit pas correctement la balise fermante, si on parenthèse mal des balises... / XML : il faut que tout soit parfait !

Le fichier XML doit être « **bien formé** ».

2. Structure d'un fichier XML

XML : souvent pour représenter une **liste / collection** d'éléments similaires

Exemple : stocker la liste des recettes que vous aimez cuisiner

Chaque recette a :

- une catégorie (entrée, plat ou dessert)
- un nom
- un ou plusieurs ingrédients (et pour chacun, une quantité)
- une ou plusieurs étapes
- un temps de préparation
- une cuisson (optionnelle) avec un temps et une température

Récupérez le fichier XML *recettes.xml* sur le site du cours et ouvrez-le avec Notepad++ pour regarder comment se présente le fichier.

On remarque qu'on a la même structure que HTML (en « arbre ») dans les fichiers XML.

La première ligne du fichier est : `<?xml version="1.0" encoding="UTF-8"?>` (elle n'est pas à connaître par coeur)

Les balises sont appelées **éléments** (par exemple "recette"), les « options » sont appelées **attributs** (par exemple "category"). Chaque élément peut avoir plusieurs éléments enfants, et aussi plusieurs attributs.

L'élément le plus « en haut » de la structure est unique, et appelé **racine**. (Ici c'est l'élément "recettes").

Il n'y a pas de règle générale pour savoir si une information doit être un élément ou un attribut : les exemples suivants donnent la même information :

```
<recette category ="plat">
  <nom> Pâtes au beurre </nom>
</recette>

<recette>
  <category> Plat </category>
  <nom> Pâtes au beurre </nom>
</recette>
```

Exercice

Créez un nouveau fichier XML qui modélise (une partie de) la liste des étudiants de l'IUT. Attention, à l'enregistrement du fichier, il faut penser à l'enregistrer sous le format XML !

Les différentes questions à se poser :

Quel est l'objet principal que je modélise ? (ça sera la racine de mon fichier)

Quel est l'objet de ma collection ? (ce sera les éléments de mon fichier)

Pour chaque objet, quelles sont ses propriétés ? (un étudiant peut avoir une date de naissance, un nom, un prénom, un numéro d'étudiant, une promo d'appartenance...)

Est-ce que je veux rajouter des catégories pour certaines propriétés ou éléments ?

Une autre approche possible serait de ne pas faire la liste des étudiants, mais la liste des promos, avec les étudiants de chaque classe comme enfants de l'élément classe.