

## Étude des langages de présentation

### Partie 1

#### Introduction

- Sauf mention contraire, le code sera à rédiger en HTML 4 avec la DTD **stricte**.
- Lorsqu'un doctype devra être modifié ou ajouté, on le copiera de préférence à partir de :  
`http://www.w3.org/QA/2002/04/valid-dtd-list.html`
- Pour la validation, nous utiliserons le validateur du W3C disponible à cette adresse :  
`http://validator.w3.org/check`
- Une archive contenant les fichiers mentionnés dans cette fiche est à récupérer sur la liste de diffusion.
- Une référence rapide pour les langages de présentation est disponible à l'adresse  
`http://http://www.w3schools.com`
- Pour les exercices faisant appel aux fonctionnalités **HTML5**, vous aurez besoin d'utiliser alternativement les navigateurs *Firefox* et *Chromium* qui ne proposent pas le même support des éléments.

#### Exercice 1 – Lecture de DTD

Récupérez les DTDs : *transitional* (`http://www.w3.org/TR/html4/loose.dtd`) et *strict* (`http://www.w3.org/TR/html4/strict.dtd`).

**Question 1.1 :** Recherchez, comparez les définitions de l'élément `body` dans les deux DTD et expliquez les différences.

**Question 1.2 :** Recherchez et comparez les définitions de l'élément `P` en expliquant les différences.

**Question 1.3 :** Décrivez la syntaxe de l'élément `head` en DTD stricte.

#### Exercice 2 – Structure d'un document HTML

**Question 2.1 :** Récupérez le fichier `toutfaux.html`. De quel type de contenu exact s'agit-il, quelle(s) ligne(s) à l'intérieur de ce document caractérise son contenu ?

**Question 2.2 :** Dessinez l'arbre syntaxique de ce document.

Consultez le court code CSS placé au début du fichier `toutfaux.html`, celui-ci affecte une couleur d'arrière plan différente aux éléments `body`, `div p` et `ul`.

**Question 2.3 :** Chargez ce fichier dans votre navigateur. Le rendu est-il conforme à ce que l'on aurait pu attendre à partir de l'arbre syntaxique ?

**Question 2.4 :** Que peut-on en déduire par rapport à l'arbre que le navigateur a construit au chargement de ce fichier ? Vous pouvez vous aider en utilisant les outils d'inspection de *Firefox* (`ctrl-shift-i`). Redessinez la portion de l'arbre correspondante.

**Question 2.5 :** Passez ce fichier dans le validateur du w3c.

**Question 2.6 :** Expliquez chaque erreur trouvée par le validateur et justifiez la en citant la DTD.

**Question 2.7 :** Proposez une correction corrigée.

**Question 2.8 :** Dessinez l'arbre syntaxique du document corrigé.

**Question 2.9 :** À partir de la version correcte, changez le doctype pour utiliser la langage XHTML 1.1 dans sa version stricte.

**Question 2.10 :** Testez d'abord le fichier dans un navigateur. Que se passe-t-il ?

**Question 2.11 :** Soumettez le au validateur ; Expliquez et corrigez à nouveau les erreurs trouvées.

#### Exercice 3 – Extensibilité de XHTML

Note : pour fonctionner correctement le lecture locale (i.e. sans passer par HTTP), vos fichiers devront porter l'extension **.xhtml** pour que le navigateur commute son parseur vers cette syntaxe, même avec un doctype correct (*cf* observations faites au TP1).

**Question 3.1 :** Créez un nouveau document en *XHTML 1.1* et ajoutez y du code en *MathML* pour y faire apparaître la formule suivante :

$$x = \sum_{i=1}^n \frac{\sqrt{b_i^2 - 4i}}{\log_a b^i}$$

Vous trouverez une référence rapide du langage ici : <http://www.mathml.su/english/>

- Vérifier le bon fonctionnement dans votre navigateur (clique droit sur la page, puis *informations sur la page*, puis vérifiez que le langage est bien **xhtml+xml+mathml**).
- Passez votre fichier au validateur.

**Question 3.2 :** Récupérez le fichier **svg\_test.txt** depuis l'archive et incorporez son contenu à votre fichier dans l'élément adéquat afin que le rendu fonctionne.

- Vérifier le bon fonctionnement dans votre navigateur (clique droit sur la page, puis *informations sur la page*, puis vérifiez que le langage est bien **xhtml+xml+mathml**).
- Passez votre fichier au validateur.
- Rappelez à quoi servent les espaces de noms (**xmlns**).
- Lisez l'explication donnée dans ce paragraphe :

<http://www.w3.org/TR/XHTMLplusMathMLplusSVG/#howto-xhtml>

et corrigez le code récupéré depuis **svg\_test.txt** en conséquence.

**Question 3.3 :** Pour voir quelque chose d'un peu plus intéressant et mesurer tout le potentiel du format SVG pour les graphismes, incorporez le contenu de **lgs\_svg.txt**. Quels sont les avantages à incorporer des graphiques sous cette forme ?

#### Exercice 4 – Questions Bonus - idées de développement

**Question 4.1 :** Écrivez en Lex/Yacc, un parseur permettant de dé-sérialiser un flux utilisant la méta-syntaxe à balises.

**Question 4.2 :** Écrivez en Lex/Yacc, un parseur permettant de dé-sérialiser une DTD (pour simplifier, on pourrait omettre les entités).

**Question 4.3 :** En se basant sur le résultat des question précédentes, écrire un programme qui valide un flux à balises en fonction d'une DTD donnée.

**Question 4.4 :** Écrire, en Java par exemple, une bibliothèque de classe permettant de modéliser un arbre HTML (en simplifiant le langage), puis, sur appel d'une simple méthode, de produire la sérialisation de cet arbre.

**Question 4.5 :** Que suffirait-il de faire pour passer en sérialisation XHTML sans avoir à tout réécrire ?