

Editeur de texte personnalisé

proposé par Ines Klimann

Projet de Programmation 2012/2013 - 51IF2IK3 - 20 septembre 2012

Ce document contient la description des deux sujets proposés. L'attribution des sujets se fait lors de la première séance. Elle apparaît ensuite sur di del et n'est plus modifiable.

1 Editeur de texte

Un *éditeur de texte* est un programme utilisé pour créer et éditer des fichiers texte. Pendant votre année de L1, vous avez pu utiliser xemacs, kedit ou un autre éditeur de texte. Souvent vous avez noté que cet éditeur ne vous convenait pas pour diverses raisons. Je vous propose ici de programmer votre propre éditeur de texte qui correspondra à vos attentes (compatibles avec celles des autres membres de votre équipe).

Attention : un éditeur de texte n'est pas un *traitement* de texte ; il ne sert pas à faire de la mise en forme.

2 But du projet

Le but de ce projet est de créer un éditeur de texte. Je donne ci-dessous la description minimale commune de ce qui doit être fait. En plus de ce fonctionnement de base, il vous est demandé d'ajouter des fonctionnalités au choix. Ces fonctionnalités seront précisées dans le cahier des charges que vous aurez à rédiger pour la semaine prochaine.

2.1 Un éditeur de base

Les fonctionnalités de base complètement indispensables :

- ouvrir un fichier existant, ouvrir un nouveau fichier,
- sauvegarder un fichier, sauvegarder un fichier sous un nouveau nom,
- couper / copier / coller.

Notez bien que vous devez fixer les modalités de ces actions (à la souris ? au clavier ? menu déroulant ? etc.). Il faut bien entendu que les textes accentués apparaissent correctement, au moins avec un type d'encodage.

Par ailleurs, pour que l'utilisation de votre logiciel soit agréable, il faut que le redimensionnement de fenêtre soit supporté.

2.2 Variante 1 : orthographe et complétion automatique

Pour ce sujet, votre éditeur de texte doit signaler les mots mal orthographiés et proposer des complétions automatiques de mots. La structure de données pour représenter un lexique (c'est-à-dire une liste de mots autorisés) est imposée : ce doit être un arbre enraciné sur le mot vide. Chaque fils d'un nœud correspond à une lettre. Chaque mot est l'étiquette d'une branche de l'arbre. Un exemple est développé figure 1.

2.3 Variante 2 : indentation html et fermeture de balises

Pour ce sujet on supposera que les sources de textes fournis sont au format HTML5 *valide*. On demande que l'éditeur indente correctement en fonction des balises HTML et qu'on puisse fermer automatiquement la bonne balise. L'indentation doit être correcte sur un document bien formé ou un début de document bien formé. On ne vous demande pas de vérifier que le document est *valide* et le comportement de votre éditeur sur un document non valide n'est pas spécifié. On vous impose d'utiliser une structure de pile pour gérer les balises HTML : une balise ouvrante s'empile, une balise fermante se dépile. Un exemple est donné figure 2 : les balises title et head ont déjà été fermées, ce qui explique qu'elles n'apparaissent plus dans la pile ; les balises DOCTYPE et meta sont sans contenu et n'ont donc jamais été empilées.

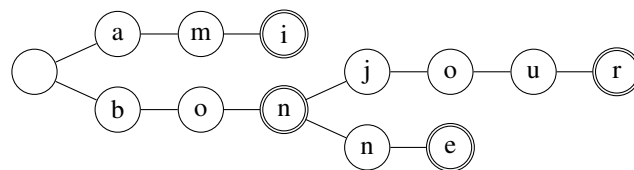


FIGURE 1 – Représentation du lexique {ami, bon, bonne, bonjour}.

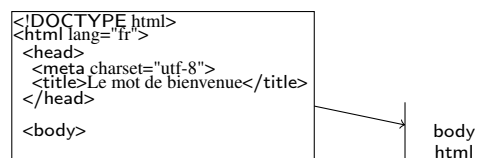


FIGURE 2 – exemple de pile de balises obtenue à partir d'un document

3 Documents à rendre

e lo22 .7400 3 . 5 m u8/memG3Tf 25/

Tout au long du semestre vous aurez à rendre les documents suivants :

- spécification fonctionnelle / cahier des charges — semaine du 24 septembre 2012,
- spécification interne — semaine du 1er octobre,
- mode d'emploi — semaine du 10 décembre,
- j'avadoc — au fur et à mesure de la production de code.

Ces documents sont décrits dans la présentation générale de l'enseignement distribuée en début de semestre et disponible sur di del :

<http://di.del.scrip.t.univ-paris-di.derot.fr/>

Après vous être connecté une première fois à <http://usvn.scrip.t.univ-paris-di.derot.fr/login> avec vos identifiant et mot de passe ENT et avoir été rattaché à votre projet, les diverses versions de votre code seront à gérer *via* le serveur svn fourni par le scrip.t :

Initialisation d'une copie locale	svn checkout --username <i>login</i> http://url
Mise à jour de la copie locale	

e l o226. 740039. 656mu8/memG/56mi 6