

## Analyse Syntaxique et Compilation, TP n° 3 : OcamlLex

Quelques pointeurs vers des docs sur `ocaml` et en particulier `ocamllex` :

- <http://caml.inria.fr/pub/docs/manual-ocaml/index.html>, ou en Français :
- <http://www.pps.jussieu.fr/Livres/ora/DA-OCAML/>

### Exercice 1 :

- Utiliser `ocamllex` pour obtenir un programme permettant de vérifier l'appariement des parenthèses ouvrantes et fermantes dans un fichier texte, et signalant le numéro de la ligne où se produit la première erreur. Extension : prendre en compte plusieurs types de "parenthèses" : `()[]{}<>`. Attention : les appariements du type `[{}]` sont interdits.
- Utiliser `ocamllex` pour obtenir un programme qui retire d'un texte en HTML tous les "tags" (commandes délimitées par `<` et `>`). Attention aux chaînes de caractères qui contiennent `>`.

### Exercice 2 :

Le but de cet exercice est d'écrire un programme d'indentation automatique de code Java à l'aide d'`ocamllex`. L'indentation consiste à ajouter des espaces au début de chaque ligne permettant une lecture plus aisée. Par exemple, voici à gauche un programme et à droite sa version modifiée.

```
class A                                     class A
{                                           {
int x;                                       int x;
void f() { x=1; }                          void f() { x=1; }
int g() {                                     int g() {
if (x>0) {                                   if (x>0) {
x--;                                         x--;
}                                             }
return x;                                    return x;
}                                             }
}                                             }
```

Nous utilisons un schéma d'indentation très simple. Nous appelons marge le nombre d'espace ajouté au début de la ligne courante. Au début, la marge vaut 0. Ensuite

- Les espaces présents au début de chaque ligne dans le code original sont ignorés et remplacés par exactement marge caractères espace.
- Si l'on rencontre un caractère `{` on l'affiche d'abord et la marge est augmentée de 4 ensuite
- Si l'on rencontre un caractère `}` on diminue la marge de 4 et on l'affiche. S'il apparaît au début d'une ligne, on diminue d'abord la marge avant de l'afficher.

On suppose que les programmes Java sont syntaxiquement correctes.

### Extensions :

- Traiter les commentaires Java qui peuvent contenir des caractères `{` ou `}`. Un commentaire Java débute par `//` et s'étend jusqu'à la fin de la ligne ou débute par `/*` et s'étend jusqu'à la première occurrence de `*/`.
- Traiter les chaînes de caractères qui sont encadrées par `"`. Elles ne peuvent contenir le caractère `"` seulement si celui-ci est précédé d'un caractère `\` et ne peuvent pas contenir de retour-chariot.