

Analyseur/Traducteur Document L^AT_EX - Document HTML

1 Sujet

Parmi ses multiples possibilités, le langage de mise en page L^AT_EX permet de réaliser des documents structurés. Le but du projet est de concevoir un traducteur qui reconnait la syntaxe de déclaration des documents L^AT_EX et de générer l'équivalent sous forme d'un document HTML.

Description succincte de L^AT_EX

Pour illustrer l'utilisation de L^AT_EX, nous allons définir un document L^AT_EX à l'aide de deux fichiers textes. Le fichier `document.tex` ci-dessous contient le préambule nécessaire à toute compilation L^AT_EX. Ce fichier restera le même quel que soit le document à produire. Le fichier `contenu.tex` ci-dessous décrit le code propre au document, c'est ce second fichier qui sera analysé par votre analyseur **Accent**.

Fichier préambule `document.tex`

```
\documentclass[a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage{a4wide}
\usepackage{times}
```

```
\input contenu.tex
```

Fichier `contenu.tex`

```
\title{Un document latex}
\author{\textit{Polytech}}
\date{}
```

```
\begin{document}
\maketitle
```

```
\section{Document \textit{Latex}}
```

```
Un document \textbf{Latex} est un document texte qui contient certaines
commandes qui seront traitées par un compilateur. \\
L'une des forces de Latex est de pouvoir générer des documents très
propres, \textit{très structurés} avec une numérotation automatique.
```

```
\subsection{Fonctionnement}
```

```
En résumé, Latex :
```

```
\begin{itemize}
\item permet d'écrire des documents à l'aide d'un simple éditeur de texte,
\item s'occupe de la mise en page,
\item offre un résultat \textit{incomparable}.
\item La réalisation d'un document se fait en 3 phases :
\begin{enumerate}
\item Edition du texte du document (vi, kwrite, xemacs, \ldots)
```

```
\item Compilation du texte (latex, pdflatex)
\item Visualisation du document généré (evince, acroread)
\end{enumerate}
\end{itemize}

\end{document}
```

La compilation et la visualisation de ce programme L^AT_EX se fait par les commandes unix suivantes :

```
pdflatex document.tex
pdflatex document.tex
evince document.pdf
```

Le fichier `pdf` généré est défini ci-dessous.

Un document latex

Polytech

1 Document *Latex*

Un document **Latex** est un document texte qui contient certaines commandes qui seront traitées par un compilateur. L'une des forces de Latex est de pouvoir générer des documents très propres, *très structurés* avec une numérotation automatique.

1.1 Fonctionnement

En résumé, Latex :

- permet d'écrire des documents à l'aide d'un simple éditeur de texte,
- s'occupe de la mise en page,
- offre un résultat *incomparable*.
- La réalisation d'un document se fait en 3 phases :
 1. Edition du texte du document (vi, kwrite, xemacs, ...)
 2. Compilation du texte (latex, pdflatex)
 3. Visualisation du document généré (evince, acroread)

Vous trouverez dans le cours **Langages et Traducteurs - GIS2A3** de la plateforme Moodle Lille 1 les fichiers `document.tex`, `contenu.tex` et `document.pdf`.

1.1 Que fait le compilateur ?

Le compilateur \LaTeX analyse le fichier source, produit un document en réalisant des opérations de mise en page en fonction des commandes placées dans le fichier source.

Si l'on regarde plus attentivement le document généré, on remarque que le compilateur a :

- généré un titre (grâce à la commande `\maketitle`) en s'appuyant sur les informations stockées à l'aide des commandes `\title`, `\author` et `\date`.
- généré automatiquement une numérotation pour les commandes `\section`, `\subsection`. Avec \LaTeX , pas besoin de se soucier d'un quelconque numéro de section, sous-section, ...
- adopté un style de police particulier pour le titre du document et les titres de section.
- généré une liste non numérotée (commande `\itemize`).
- généré une liste automatiquement numérotée (commande `\enumerate`).
- réalisé une mise en gras (commande `\textbf`) et une mise en italique (commande `\textit`) de certaines zones de texte.

1.2 La syntaxe \LaTeX

La syntaxe \LaTeX est très simple, toute commande commence par le caractère `\`. On peut distinguer cependant trois catégories de syntaxe de commande :

- 1. Les commandes simples sans paramètres. Le caractère `\` est suivi du nom de la commande.

Syntaxe : `\nomCommande`

- 2. Les commandes dont l'action s'applique dans une zone définie par des accolades.

Syntaxe : `\nomCommande{zone...}`

- 3. Les commandes dont l'action s'applique dans une zone définie par des commandes de début et de fin.

Syntaxe : `\begin{nomCommande} zone ... \end{nomCommande}`

Nom Commande	Catégorie	Désignation
document	3	délimite un document \LaTeX
title	2	définit le titre du document
author	2	définit l'auteur du document
date	2	définit la date du document
maketitle	1	génère un titre
section	2	définit un titre de niveau 1
subsection	2	définit un titre de niveau 2
subsubsection	2	définit un titre de niveau 3
textbf	2	définit une zone en gras
textit	2	définit une zone en italique
itemize	3	définit une zone de liste non numérotée
enumerate	3	définit une zone de liste numérotée
item	1	définit un élément d'une liste
\	1	passage à une nouvelle ligne
ldots	1	points de suspension

Syntaxiquement, un document \LaTeX peut être considéré comme une suite de mots et de commandes. Un mot peut contenir des lettres minuscules et majuscules, des chiffres, des lettres accentués et des caractères spéciaux parmi l'ensemble { : , ; . - ' () }.

Les commandes \LaTeX doivent respecter les contraintes suivantes :

- A l'exception des commandes optionnelles `\title`, `\author` et `\date`, toutes les autres commandes doivent se trouver à l'intérieur de la commande unique et obligatoire `\document`.
- Les commandes `\title`, `\author` et `\date` peuvent contenir des mots et les commandes `\textit`, `\textbf` et `\ldots`.
- Les commandes `\title`, `\author` et `\date` peuvent être introduite dans n'importe quel ordre.
- Les commandes `\textit` et `\textbf` peuvent s'imbriquer et peuvent contenir des mots et la commande `\ldots`.
- La commande `\maketitle` est optionnelle et doit apparaitre une seule fois juste après la commande `\begin{document}`.

- Les commandes `\section`, `\subsection` et `\subsubsection` ne peuvent s'imbriquer.
- Les commandes `\section`, `\subsection` et `\subsubsection` peuvent contenir des mots et les commandes `\textit`, `\textbf` et `\ldots`.
- Les commandes `\section`, `\subsection` et `\subsubsection` ne peuvent apparaître dans une zone de liste.
- Un `\begin{itemize}` ou un `\begin{enumerate}` est suivi soit d'un `\item`, soit d'un `\begin{itemize}`, soit d'un `\begin{enumerate}`.
- La commande `\item` ne peut apparaître que dans une zone de liste.
- Les espaces, tabulations et retour-chariot sont ignorés par le compilateur \LaTeX .

Remarque : Certaines des contraintes ci-dessus ont été introduites pour simplifier le sujet. Le compilateur \LaTeX est plus permissif.

2 Découpage du projet

Le projet est décliné en plusieurs versions. Vous devez implémenter chaque version à l'aide des outils `Lex` et `Accent`.

Attention, la grammaire définie dans la première version ne doit pas être modifiée dans les versions suivantes. Seuls des attributs et des actions sémantiques pourront être ajoutés à cette version de base.

2.1 Première version : définition de la grammaire

En se basant sur les règles syntaxiques définies dans la section 1.2, écrivez un analyseur qui accepte des documents \LaTeX . Vous pouvez tester votre version avec le fichier `contenu.tex` téléchargeable sur la plateforme Moodle Lille 1.

2.2 Deuxième version : vérification de la cohérence

A partir de la grammaire définie dans la **première version**, proposez un schéma de traduction dirigé par la syntaxe permettant de vérifier que :

- Les commandes `\title`, `\author` et `\date` ne sont définies qu'une seule fois au plus.
- La commande `\subsection` ne peut être définie que si une commande `\section` a été préalablement définie.
- La commande `\subsubsection` ne peut être définie que si une commande `\subsection` a été préalablement définie dans la section courante.

Il est à noter que le compilateur \LaTeX est plus permissif que votre analyseur et accepte ainsi les deux dernières règles.

2.3 Troisième version : génération HTML

Dans cette version, vous allez enrichir la version précédente pour générer un fichier HTML qui aura une forme équivalente au document \LaTeX généré. En particulier, vous utiliserez les balises HTML suivantes :

- `<meta charset="UTF-8"/>` pour déclarer l'encodage du fichier HTML généré.
- `H1`, `H2` et `H3` pour définir les zones de titres. Vous devez effectuer la numérotation des sections.
- `TITLE` pour définir le titre du document. Vous afficherez **le nom de l'auteur** en paramètre si celui-ci a été spécifié.
- `UL` et `OL` pour définir des listes.

Remarques :

- Pour faciliter les traitements, vous pouvez :
 - Utiliser des variables globales pour la gestion de la numérotation des sections.
 - Utiliser une variable globale de type vecteur de caractères et les fonctions de la bibliothèque `string.h` pour stocker une partie du texte avant de l'afficher.
- Si vous devez utiliser un attribut d'un non-terminal de type chaîne de caractères, vous devez déclarer un type `chaîne` de type `char *` (`typedef char * chaîne;`) et indiquer que cet attribut est de type `chaîne`.

La figure ci-dessous est une copie écran du document HTML généré à partir de notre exemple.



3 Références

Vous trouverez sur le web de multiples références concernant L^AT_EX et HTML. Voici deux références possibles :

<http://www.commentcamarche.net/contents/498-html-langage>
http://www-igm.univ-mlv.fr/~dr/XPOSE2004/latex/bases_elementaires.html

4 encodage

Pour éviter tout problème de compatibilité, tous vos fichiers doivent être encodés en UTF-8.

5 Travaux à rendre

5.1 Travaux intermédiaires

A la fin de chaque séance, vous devez déposer sur la plateforme Moodle Lille 1, l'ensemble des travaux que vous avez réalisés depuis le début du projet en suivant les consignes suivantes :

1. Créez une archive compressée des documents à déposer avec la commande :

```
tar -czvf nom1_nom2.tgz Dossier
```

où `nom1` et `nom2` sont les noms des étudiants composant votre groupe et `Dossier` est le répertoire contenant les documents que vous devez rendre.

2. Sélectionnez dans le cours **Langages et Traducteurs - GIS2A3** la séance appropriée.
3. Déposez votre archive compressée.

5.2 Rendu final

Vous devez déposer pour le **vendredi 21 juin 16h** au plus tard :

- Sur la plateforme Moodle Lille 1, une archive de vos versions contenant des fichiers de test pour chaque version (les fichiers de vos différentes versions doivent être rangés dans des répertoires de noms V1, V2 et V3).
- Un rapport imprimé dans le casier courrier de votre tuteur.

Votre rapport doit être décomposé en fonction des versions à réaliser. Vous devez décrire dans ce rapport :

- les grammaires et schémas de traduction dirigée par la syntaxe utilisés,
- les structures et types que vous avez définis,
- les attributs et variables utilisés, et leurs objectifs,
- les tests que vous avez réalisés,
- ...

La conclusion de votre rapport doit comporter une critique du travail réalisé et mentionner des perspectives d'amélioration de votre travail.

Votre rapport doit contenir également une impression de vos différents fichiers.