

Introduction à \LaTeX

Organisation et automatisation sous \LaTeX

Pascal Bessonneau
DEPP

03/2013

Plan

Organiser un document

Définir une fonction personnalisée

Organiser un document L^AT_EX

Comme les images et les tableaux peuvent être contenus dans des dossiers séparés, il est conseillé de mettre les images dans un sous-répertoire à part et les tableaux dans un autre sous-répertoire. Lors de l'inclusion, il suffit alors d'ajouter le nom du sous-répertoire et le nom du fichier sans le suffixe.

Organiser un document \LaTeX

Code

```
\begin{figure}
  \includegraphics[scale=1]{Graphiques/calvin_hobbes.jpg}
  \caption{Calvin et Hobbes}
  \label{Graphiques_Calvin}
\end{figure}
```

Les chemins sous \LaTeX

Dans les chemins (noms de fichiers et répertoires), \LaTeX ne supporte pas les caractères spéciaux et notamment les blancs. Quand vous spécifiez le chemin pour un style, une image, un tableau, une partie de document, vous ne pouvez donc pas utiliser de caractères spéciaux accentués et de blancs dans les chemins.

L'inclusion dans un document volumineux

Vous pouvez inclure du code \LaTeX provenant d'un autre fichier dans un document \LaTeX .

Cela permet, par exemple, de faire un livre avec un fichier par chapitres ou bien de ne pas inclure les tableaux directement dans le corps du document (avec Sweave par exemple).

L'inclusion dans un document volumineux

Pour cela vous disposez de deux commandes *input* et *include*. La commande *input* insère directement le code \LaTeX tandis que la commande *input* insère un saut de page avant le contenu du document inséré.

Toutefois on ne peut pas imbriquer un appel à un fichier à l'intérieur d'un autre. On ne pourra donc pas inclure un tableau par un *include* dans un fichier qui sera lui-même importé par un autre *include*.

Définir une fonction personnalisée


Pour insérer un graphique dans un document de façon répétée, autant l'écrire une fois... dans le préambule...

Code

```
\newcommand{%  
\LinuxA}  
{\includegraphics[scale=0.1]  
{Graphiques/linux.png}}
```


Définir une fonction personnalisée

Dans cet exemple, on définit une commande `\Linux` qui ajoute une image.

 est mon OS préféré.

Mais on voit qu'il y a un problème car le blanc entre le texte et la macro disparaît.

xspace

La même fonction avec l'utilisation du package *xspace*.

Code

```
\newcommand{%  
\LinuxB}  
{\includegraphics[scale=0.1]  
{Graphiques/linux.png}\xspace}
```

xspace



est mon OS préféré.

Le `\xspace` situé en fin de commande va, lors de la compilation, vérifier si un espace est nécessaire et l'ajouter si besoin.

Avec des arguments

On aurait pu écrire une fonction plus générique comme par exemple :

Code

```
\newcommand{%  
\Image}[2]  
{\includegraphics[scale=#1]  
{Graphiques/#2}}
```

Dans ce cas l'appel est le suivant :

Code

```
\Image{0.1}{linux.png}
```

Redéfinition d'une fonction existante

On pourrait vouloir par exemple modifier une fonction existante pour les besoins de la mise en page. La redefinition utilise une syntaxe légèrement différente.

La fonction ci-dessous remplace l'italique de la fonction *emph* par des caractères en gras.

Code

```
\renewcommand{%  
\emph}[1]  
{\textbf{#1}}
```

Package de programmation

Enfin pour faire de la programmation simple, on peut créer des variables en \LaTeX et les packages *ifthen* permet de faire des boucles et des tests conditionnels.