



# Initiation à L<sup>A</sup>T<sub>E</sub>X

## Bibliographies, commandes, paquetages

Pierre Senellart



Semaine *Informatique pratique*, septembre 2022



# Plan

Bibliographies

BibTeX

Citations

Commandes

Paquetages et classes

Mais aussi...



# Plan

Bibliographies

Bib $\text{T}_{\text{E}}\text{X}$

Citations

Commandes

Paquetages et classes

Mais aussi...



## BibT<sub>E</sub>X

- Logiciel de gestion de **bibliographies**
- Document BibT<sub>E</sub>X : fichier texte **.bib** décrivant des références bibliographiques, avec une syntaxe spéciale.
- Chaque référence est identifiée par une **clef** alphanumérique.
- Principe général : on compile une fois avec **pdflatex** pour récolter les citations bibliographiques, on appelle **bibtex** sur le nom du document L<sup>A</sup>T<sub>E</sub>X sans le **.tex**, puis on recompile deux fois avec **pdflatex** pour intégrer les références bibliographiques.
- Vieux logiciel : parfois des petits côtés pénibles pour le faire marcher. En particulier, il faut parfois encoder les accents « à l'ancienne » (p. ex., `\'e` pour « é »), protéger les macros avec des accolades. . .
- **biber** : logiciel plus moderne, qui peut remplacer **bibtex** en particulier quand on utilise le package **biblatex**



## Fichiers auxiliaires

- À partir des documents (`.tex`), paquetages (`.sty`), classe (`.cls`), etc.,  $\LaTeX$  produit le document final (`.pdf` ou `.dvi`) ainsi que des fichiers auxiliaires :
  - `.aux` toutes les références rencontrées dans le fichier
  - `.toc` les titres qui constitueront la table des matières
  - `.out` les bookmarks générés par hyperref
  - `.log` un log de compilation
  - etc.
- À partir du `.aux`, du fichier de références (`.bib`), du style de bibliographie (`.bst`), Bib $\TeX$  produit un fichier `.bb1` qui contient la bibliographie formatée.
- Si un tel fichier `.bb1` existe, il est utilisé pour produire le document final à la prochaine compilation.



## Exemple de référence BibTeX

```
@article{ bryant92symbolic,  
  author = "Randal E. Bryant",  
  title = "Symbolic {Boolean} Manipulation with  
Ordered Binary-Decision Diagrams",  
  journal = {ACM Computing Surveys},  
  volume = 24,  
  number = 3,  
  pages = {293-318},  
  year = 1992}
```

Un fichier BibTeX est une suite de telles références.

bryant92symbolic est la clef, les autres lignes décrivent les champs; la valeur d'un champ complexe est mis entre guillemets droites ou entre accolades.



## Types de références les plus courants

@article article de journal

@book livre

@incollection chapitre de livre

@inproceedings article publié dans les actes d'une conférence

@mastersthesis thèse de Master

@misc divers

@phdthesis thèse de doctorat

@techreport rapport technique

@unpublished non publié



## Champs existants

**author** auteurs du document, doivent être séparés par des « and »  
(p. ex., "Jean Dupont and Jacques Durand" ou  
"Dupont, Jean and Durand, Jacques")

**title** titre du document ; si celui-ci contient des majuscules qui  
doivent impérativement être présente (acronyme, nom  
propre), doivent être protégées par des accolades

**month** mois, sous forme d'abréviation : jan, feb, mar, apr, may,  
jun, jul, aug, sep, oct, nov, dec

**institution, school** pour les rapports techniques ou les thèses

**booktitle** pour les articles de conférences et les chapitres de livre

**editor** pour l'éditeur d'un ouvrage collectif

**publisher** pour la maison d'édition d'un livre

**address** pour l'adresse de l'éditeur

**volume, number, pages, year, note**



# Plan

## Bibliographies

Bib $\text{T}_{\text{E}}\text{X}$

Citations

Commandes

Paquetages et classes

Mais aussi...



## En L<sup>A</sup>T<sub>E</sub>X

```
D'après~\cite{bryant92symbolic},  
les OBDD sont très intéressants.
```

```
...
```

```
\bibliographystyle{alpha}  
\bibliography{biblio}
```

- `\cite{clef}` pour citer une référence, `\nocite{clef}` pour inclure une référence dans la bibliographie sans la citer, `\nocite{*}` pour inclure toutes les références
- Styles de bibliographie standard : alpha, plain, unsrt
- **biblio** désigne le nom du fichier BibT<sub>E</sub>X (sans l'extension)



## Personnaliser les références

- Nombreux styles de références, y compris des styles français (unsrt-fr, plain-fr, alpha-fr, etc.)
- paquetages pour personnaliser les citations dans le corps du document : **natbib**, **jurabib**
- paquetages pour avoir des bibliographies multiples : **bibtopic**, **multibib**, **splitbib**
- créer ses propres styles bibliographiques : difficiles avec le système classique, mais un paquetage comme **biblatex** facilite les choses



# Plan

Bibliographies

Commandes

Définir des commandes  $\text{\LaTeX}$

Définir des commandes en  $\text{\TeX}$

Paquetages et classes

Mais aussi...



# Plan

Bibliographies

Commandes

Définir des commandes  $\text{\LaTeX}$

Définir des commandes en  $\text{\TeX}$

Paquetages et classes

Mais aussi...



## Commandes sans arguments

```
\newcommand{\hello}  
{Bonjour, ça va?\par}  
  
\hello\hello\hello
```

Bonjour, ça va ?

Bonjour, ça va ?

Bonjour, ça va ?



## Redéfinir une commande

```
\[\epsilon\leq\phi\geq\emptyset\]
```

```
\renewcommand{\epsilon}{\varepsilon}
```

```
\renewcommand{\phi}{\varphi}
```

```
\renewcommand{\emptyset}{\varnothing}
```

```
\renewcommand{\leq}{\leqslant}
```

```
\renewcommand{\geq}{\geqslant}
```

```
\[\epsilon\leq\phi\geq\emptyset\]
```

$$\epsilon \leq \phi \geq \emptyset$$

$$\varepsilon \leq \varphi \geq \varnothing$$



## Commandes avec arguments

```
\newcommand{\textitbf}[1]
{\textit{\bfseries #1}}
```

C'est `\textitbf{gras  
italique}`.

C'est ***gras italique***.

Il y a aussi `\newcommand*` qui impose que son argument ne comporte pas de saut de paragraphe.

## Commandes avec argument optionnel

```
\newcommand{\monsieur}[2] [M.]  
{#1~\textsc{#2}}
```

```
\monsieur[Jean]{Dupont} et  
\monsieur{Durand} sont venus.
```

Jean DUPONT et M. DU-  
RAND sont venus.

On donne la valeur par défaut de l'argument optionnel (il ne peut y en avoir qu'un) après le nombre d'arguments.

## Environnements

```
\newenvironment{listeunel}  
{\begin{itemize}\item }  
{\end{itemize}}  
  
\begin{listeunel}  
toto  
\end{listeunel}
```

- toto

On peut aussi définir des environnements avec arguments, avec argument optionnel, redéfinir des environnements, etc.



# Plan

Bibliographies

**Commandes**

Définir des commandes  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

Définir des commandes en  $\text{T}_{\text{E}}\text{X}$

Paquetages et classes

Mais aussi...



## Alias

```

\let\t\textbf
\renewcommand{\textbf}[1]
  {\textit{#1}}

\t{toto} \t{titi}

\textbf{toto} \textbf{titi}

```

**toto titi**  
*toto titi*



## Définition de macro T<sub>E</sub>X

```
\def\bouh#1(#2)+#3
{\textbf{#1}\textit{#2}%
\textsc{#3}}
```

**toti**TA

```
\bouh to(ti)+ta
```

Définition d'une macro avec syntaxe (à peu près) arbitraire !



# Plan

Bibliographies

Commandes

**Paquetages et classes**

Créer un paquetage

Créer une classe

Installation

Mais aussi...



# Plan

Bibliographies

Commandes

**Paquetages et classes**

Créer un paquetage

Créer une classe

Installation

Mais aussi...



## Principes généraux

- paquetage : fichier L<sup>A</sup>T<sub>E</sub>X avec extension `.sty`
- Dans un paquetage, on utilise `\RequirePackage` plutôt que `\usepackage`
- Seule différence importante : dans un paquetage, des noms de commandes peuvent contenir un caractère `@`, pas dans un document. On réserve donc les noms de commande avec `@` à des commandes internes.
- Au cas où on souhaite quand même utiliser une commande avec `@` dans un document on peut utiliser `\makeatletter` avant et `\makeatother`
- Possibilité de gérer des options de paquetage



## Squelette d'un paquetage

```

\NeedsTeXFormat{LaTeX2e}

\ProvidesPackage{mypackage} % Nom du paquetage

\DeclareOption*{
  \PackageError{mypackage}
    {Unknown option `\'CurrentOption'}
}
\DeclareOption{optiona}{codea}
\DeclareOption{optionb}{codeb}
\ProcessOptions

% Code arbitraire

```

codea est exécuté si l'option optiona est fournie.



## Faciliter le traitement des options

```
\newif\if@optiona
\DeclareOption{optiona}{\@optionatrue}

\ProcessOptions

\if@optiona
...
\else
...
\fi
```



# Plan

Bibliographies

Commandes

**Paquetages et classes**

Créer un paquetage

**Créer une classe**

Installation

Mais aussi...



## Classe de document

- Très similaire à un paquetage, fichier **.cls**
- En général, construit à partir d'une classe de base
- Squelette :

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{classe} % Nom de la classe

% On peut passer des options à la classe de base
\PassOptionsToClass{options}{classebase}

% On peut traiter des options comme pour les
% paquetages avec \DeclareOption, \ProcessOptions

% On charge la classe de base
\LoadClass{classebase}

% Autres instructions
```



# Plan

Bibliographies

Commandes

Paquetages et classes

Créer un paquetage

Créer une classe

**Installation**

Mais aussi...



## Installation d'un paquetage ou d'une classe

- Si c'est juste un fichier : le copier dans le répertoire de compilation
- Sinon, plus complexe. Sous Unix :
  - Créer un répertoire `$HOME/texmf/tex/latex/`
  - Y placer l'ensemble des fichiers du paquetage (décompressés), dans un sous-répertoire au nom du paquetage
  - Taper `texhash ~/texmf` en ligne de commande
  - Tester !
  - Il y a aussi la variable d'environnement `TEXINPUTS` qui spécifie des répertoires d'où les fichiers d'entrée sont chargés



# Plan

Bibliographies

Commandes

Paquetages et classes

Mais aussi...



## Autres fonctionnalités de L<sup>A</sup>T<sub>E</sub>X

- créer des index avec **makeindex**, un programme externe similaire à BibT<sub>E</sub>X
- incorporer du code source avec le paquetage **minted**

```

\begin{minted}{c}
int main(void) {
    printf("Hello.\n");
    return 0;
}
\end{minted}

```

```

int main(void) {
    printf("Hello.\n");
    return 0;
}

```

- définir des algorithmes en pseudo-code avec les paquetages **algorithmicx** ou **algorithm2e**
- et bien d'autres choses...