



# Outils informatiques pour la communication

Céline Alec

Université de Caen-Normandie

celine alec@unicaen.fr

#### Contenu

- Modéliser des diagrammes de classes UML
- Écrire avec LATEX
  - des rapports
    - bases d'un rapport
    - citer ses sources avec BibTeX
    - citer ses algorithmes avec les paquetages algorithm2e et listings
    - faire des figures avec le paquetage tikz
  - des diaporamas de soutenance (classe beamer)

# Organisation du cours

- Format:
  - 2h CM
  - 10h TP
- Modalités de contrôle des connaissances :
  - 100 % CC
  - 2 notes
  - Pas de session 2
  - Note finale : Max( (CC1+CC2)/2; CC2 )

### **UML**

#### Définition

Le langage de modélisation UML (Unified Modeling Language) est un langage graphique permettant de représenter des modèles de systèmes utilisant des objets.

Un modèle est une représentation abstraite et simplifiée (i.e., qui exclut certains détails), d'une entité (phénomène, processus, système...) du monde réel en vue de le décrire, de l'expliquer ou de le prévoir.

Un objet est une entité qui regroupe :

- des attributs (qui définissent son être)
- des méthodes (qui définissent son comportement)

### **UML**

UML dispose de plusieurs types de modèles, appelés diagrammes, chaque diagramme représentant une vue distincte du système.

Il existe 2 groupes de diagrammes :

- les diagrammes structurels (diagrammes de classes, de paquetages, de composants, etc.) :
- les diagrammes comportementaux (diagrammes de cas d'utilisation, d'états-transitions, de séquences, etc.)

lci, nous nous focaliserons uniquement sur les diagrammes de classes et de paquetages.

# Diagrammes de classes

#### Définition

Le diagramme de classes est un diagramme structurel qui permet de représenter :

- des classes (attributs et méthodes)
- des associations (relations) entre les classes

### Une classe

Une classe est représentée par un rectangle divisé en 3 compartiments.

- Le premier compartiment contient le nom de la classe (qui peut être aussi une classe abstraite ou une interface)
- Le deuxième compartiment contient les attributs
- Le troisième compartiment contient les méthodes

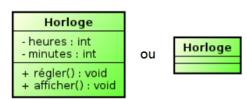
```
Nom DeLaClasse

-nomAttribut1
-nomAttribut2: type
-nomAttribut3: type = valeur

+nomOperation1()
#nomOperation3(parametre1)
-nomOperation3(parametre2: type, parametre3: type)
#nomOperation4(): typeRetour
-nomOperation5(parametre2: type, parametre3: type): typeRetour2
```

Si la modélisation ne s'intéresse qu'aux relations entre les différentes classes du système (et pas au contenu des classes), nous pouvons ne pas représenter les attributs et/ou les méthodes de chaque classe.

# Exemple



#### Détails

On peut indiquer si besoin :

- la visibilité des méthodes et des attributs grâce aux symboles + (public), (private), # (protected),  $\sim$  (package)
- le type de chaque attribut
- la signature de chaque méthode
- la valeur par défaut des attributs et paramètres de méthodes
- les attributs et méthodes de classe (statiques), représentés par un nom souligné
- le type spécifique de la classe (interface, classe abstraite, énumération)

La syntaxe peut varier suivant l'outil utilisé.

#### Les relations entre les classes

Les classes sont liées entre elles par différents types de relations :

- l'association (la relation la plus standard)
- l'héritage
- la composition
- l'agrégation

## L'association

Elle spécifie une liaison entre deux classes.

Personne				Entreprise
nom: String prénom: String	employé *	travailler pour	employeur 1	raison sociale: String
salaire(): Float				chiffre_d_affaires(): Float

#### Il peut y figurer :

- son nom
- un sens de lecture (symbolisé par une flèche)
- une multiplicité à chaque extrémité : indiquant le nombre d'instances impliquées dans la relation
- un rôle à chaque extrémité : il s'agit d'un nom qui indique la façon dont l'objet est vu de l'autre côté de l'association

# Les multiplicités

Un employé travaille pour une et une seule entreprise. Une entreprise contient 0 ou plusieurs employés.

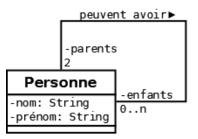
Personne				Entreprise
nom: String prénom: String	employé *	travailler pour	employeur 1	raison sociale: String
salaire(): Float				chiffre_d_affaires(): Float

#### On peut utiliser :

- 0..1
  - 1..1 ou 1
  - 0..\* ou 0..n ou \*
  - 1..\* ou 1..n
  - éventuellement des nombres bien précis (4..5, 10, etc.)

# Cas particulier d'association

L'association réflexive lie une classe avec elle-même.



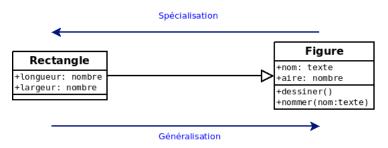
## Classe particulière : l'énumération

L'énumération est une classe particulière, souvent utilisée pour définir un ensemble fini de valeurs.

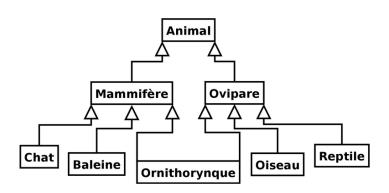


# L'héritage

L'héritage est à l'image de l'héritage dans les langages de programmation objet. Il se matérialise par une flèche pointant sur la classe mère.



### Exemple



Intro Diagrammes LATEX Conclusion

# La composition et l'agrégation

Il s'agit de cas particuliers d'association. Elles sont utilisées dans le cas où une classe A est constituée d'une classe B. Dans ce cas, on dispose un losange du côté de la classe A.

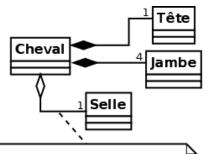
Dans le cas d'une composition, la classe B ne peut pas exister indépendamment de la classe A. Si une instance de la classe A disparaît, alors son instance de la classe B disparaît aussi.

Dans le cas d'une agrégation, les classes sont indépendantes. La suppression d'une instance de la classe A n'entraîne pas la suppression de son instance associée de la classe B.

Dans le cas d'une composition, le losange est plein. Dans celui d'une agrégation, le losange est vide.

| Intro | Diagrammes | LATEX | Conclusion | October | O

### Exemple



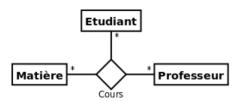
La selle ne fait pas réellement partie du cheval. Elle peut être utilisée pour un autre cheval.

ntro Diagrammes LATEX Conclusion

### Les associations n-aires

Une association n-aire lie plus de 2 classes entre elles. Elle est représentée par un losange lié à toutes les classes concernées.

La multiplicité associée à une classe A représente le nombre d'instances de A pour une instance de chacune des autres classes. Par exemple, soient les classes A, B, C. La multiplicité associée à A sera associée à la question suivante : « Pour un B et un C, combien a-t-on de A? »

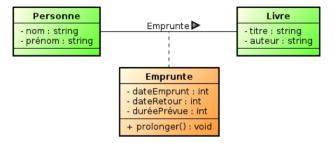


On peut choisir de créer une classe supplémentaire à la place du rectangle, permettant de la traduire en un ensemble d'association binaire. Dans ce cas, il faut revoir les multiplicités.

Intro Diagrammes ATEX Conclusion

### Classe d'association

Une association peut apporter de nouvelles informations (attributs et méthodes) qui n'appartiennent à aucune des deux classes qu'elle relie et qui sont spécifiques à l'association. Ces nouvelles informations peuvent être représentées par une nouvelle classe attachée à l'association via un trait en pointillés.

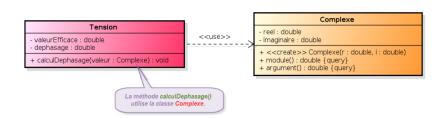


lci aussi, on peut convertir la classe d'association en une classe standard et des associations binaires en revoyant les multiplicités.

## Relation de dépendance

La dépendance (« use ») permet d'exprimer qu'une classe utilise l'autre. Elle est habituellement utilisée lorsqu'une classe utilise un objet d'une autre classe comme argument dans la signature d'une méthode ou alors lorsque l'objet de l'autre classe est créé à l'intérieur de la méthode.

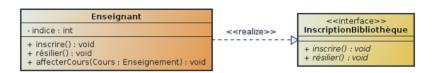
Elle est représentée par un trait discontinu orienté reliant les deux classes et éventuellement l'expression  $\ll$  use  $\gg$ .



### Relation de réalisation

La réalisation (« realize ») permet d'exprimer qu'une classe implémente une interface.

Elle est représentée par un trait discontinu avec une flèche triangulaire et éventuellement l'expression « realize ».



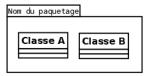
# Diagrammes de paquetages

Le diagramme de paquetages est un diagramme qui représente les paquetages composant un système, ainsi que les relations qui lient ces différents paquetages.

Un paquetage est représenté par un dossier.

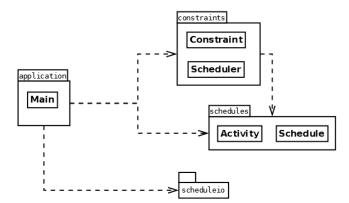


Les classes d'un paquetage peuvent être représentées à l'intérieur du dossier.



# Diagrammes de paquetages

Il y a une relation de dépendance (« use ») entre deux paquetages lorsqu'au moins une classe d'un paquetage a besoin d'utiliser une classe de l'autre paquetage.



o Diagrammes LATEX Conclusion

# Diagrammes: conclusion

Les diagrammes de classes et de paquetages ont pour but de modéliser un système informatique.

Lors de la conception, il est important d'avoir conscience du but de la modélisation (générer du code automatiquement, expliquer à quelqu'un ce qu'on représente, etc.). Suivant cela, on ne va pas s'intéresser aux mêmes points de détails.

Il faut faciliter la compréhension du lecteur en ne chargeant pas trop les diagrammes. Dans un rapport de projet informatique, il est plutôt conseillé de faire un diagramme de paquetages pour montrer le projet d'une manière générale, puis un diagramme de classes pour chaque paquetage.

Un diagramme aide à la compréhension mais ne remplace pas du texte explicatif.

LATEX est un langage créé pour séparer le fond de la forme lors de la création d'un document.

Les fichiers LATEX sont décrits dans un fichier .tex. Ce fichier doit être « compilé » afin d'être transformé en un fichier imprimable (.pdf en général).

# ATEX.

Pour utiliser LATEX, on a besoin de:

- Une distribution LaTeX: logiciel comportant toutes les composantes de LaTeX, transforme via LaTeX votre code tapé dans l'éditeur LaTeX en un document PDF. Ex: MiKTeX. TeX Live.
- Un éditeur LaTeX : c'est un logiciel vous permettant de taper votre texte, de mettre en couleur les différents codes utilisés, de numéroter vos lignes, etc.
  - Choisir un éditeur LaTeX vous simplifie la vie, mais il n'est pas indispensable : il est possible de taper du LATEX dans un simple éditeur de texte (même si cela est moins pratique). Il propose aussi bon nombre de raccourcis et boutons utiles à la création de document. (Ces boutons et raccourcis diffèrent d'un éditeur à l'autre).

# Démonstration simple

Soit le fichier test tex avec le contenu suivant :

```
\documentclass[12pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\begin{document}
blabla
\end{document}
```

#### Exécutons :

pdflatex test.tex

# Utilisation d'un éditeur LATEX

LATEX est utilisable sans problème avec une distribution LATEX et un éditeur de texte basique.

On peut néanmoins utiliser un éditeur La TEX, c'est-à-dire un éditeur de texte contenant des astuces rapides pour rédiger et compiler des commandes LATEX.

Des exemples d'éditeurs sont donnés sur https://www.gutenberg.eu.org/Editeurs-pour-La-TeX.

## Exemples d'éditeurs LATEX

#### Texmaker



#### TeXworks



#### TeXstudio



#### Overleaf



tro Diagrammes LATEX Conclusion

# Commandes principales

```
\documentclass[12 pt]{ article}
\usepackage [utf8]{inputenc}
\usepackage [T1]{fontenc}
\usepackage [french]{ babel}
\begin {document}
blabla
\end{document}
```

- La commande \documentclass[options] {type} exprime dans ses accolades le type du document (article, book, letter, report). Elle se place au début et peut avoir des options entre crochets.
- La commande \usepackage[option] {type} se place juste après.
   Elle permet d'exploiter un « package » particulier. Ici, les trois commandes mentionnées permettent de définir l'encodage des caractères, l'encodage de la police et la langue du document.
- Après ces commandes, un document LaTeX commence toujours par \begin{document} et finit par \end{document}.

Intro Diagrammes LATEX Conclusion

#### Les éléments de structure

Éléments	Commandes
Partie	\part{nom de la partie}
Chapitre <sup>1</sup>	\chapter{nom du chapitre}
Section	\section{nom de la section}
Sous-section	\subsection{nom de la sous section}
Sous-sous-section	\subsubsection{nom de la sous sous section}
Paragraphe	\paragraph{nom du paragraphe}
Sous-paragraphe	\subparagraph{nom du sous paragraphe}

<sup>1.</sup> Pas disponible pour les types article et letter

## Quelques commandes utiles

Pour mettre un commentaire dans son code on utilise %

```
\begin{document}
Bla Bla%Seulement Bla Bla va s'afficher
\end{document}
```

- La table des matières peut être créée automatiquement avec \tableofcontents.
- Pour mettre du texte en italique : \textit{texte}.
- Pour mettre du texte en gras : \textbf{texte}.
- Pour insérer une note de bas de page : \footnote{contenu de la note de bas de page}.

# Page de garde

On peut créer une page de garde basique avec trois éléments :

- le titre du document (commande \title{votre titre});
- l'auteur (commande \author{les noms des auteurs});
- la date (commande \date{la date que vous souhaitez}).

Ces trois éléments sont introduits avant la commande \begin{document}, et une quatrième commande \maketitle se place juste après \begin{document} afin de faire comprendre à LaTeX que vous souhaitez composer une page de garde avec les trois éléments cités plus haut :

```
\title{Mon premier document {\LaTeX}}
\author{C. Alec}
\date{}

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

#### Listes

Pour créer une liste à puces :

- ouverture de l'environnement itemize;
- écriture de chaque élément de la liste précédé de la commande \item;
- fermeture de l'environnement itemize.

```
\begin{itemize}
  \item nom1
  \item nom2
\end{itemize}
```

- o nom1
- nom2

Pour une liste numérotée, on fait de même, mais avec l'environnement enumerate.

```
\begin{enumerate}
  \item nom1
  \item nom2
\end{enumerate}
```

- $lue{0}$  nom1
- nom2

### Alignement

• Pour gérer l'alignement :

```
\begin{flushleft}
texte à gauche
\end{flushleft}
\left|
\texte \texten \text
```

texte à gauche

texte à droite

texte centré

- Pour faire un saut de page, on utiliser la commande \newpage.
- Pour que la page de garde n'ait pas de numéro de page et que la page suivante commence à la page 1, on place dans la page de garde :

```
\thispagestyle{empty}
\setcounter{page}{0}
```

### Références internes

Pour les références internes, 3 commandes :

- \label{nom\_choisi} sert à marquer un endroit
- \pageref{nom\_choisi} et \ref{nom\_choisi} permettent d'appeler le numéro de page ou la référence de l'élément marqué dans une autre zone de votre document.

```
La section \ref{sec:blabla}
introduit blabla.
Elle se situe page \pageref{sec:
blabla}.
\section{Bonjour}
Bonjour
\section{Blabla}
\label{sec:blabla}
blabla
```

# Cet exemple permet d'obtenir un texte de la forme suivante :

La section 2 introduit blabla. Elle se situe page 1.

#### 1 Bonjour

Bonjour

#### 2 Blabla

blabla

# Hyperliens

Pour générer un lien vers un site web, on peut utiliser la commande \url{URL} ou la commande \href{URL}{texte affiché cliquable}. Ces commandes ne fonctionnent que si l'on a chargé le package hyperref au préalable.

```
cas 1 : \href{https://www.google.com/}{Lien vers Google}
cas 2 : \url{https://www.google.com/}
```

```
cas 1 : Lien vers Google
cas 2 : https://www.google.com/
```

# Les flottants

Les « flottants » sont l'ensemble des éléments qui perturbent le flot du texte et désignent classiquement les figures et tableaux.

Pour placer une figure, on utilise l'environnement figure.

```
\begin{figure}[option de placement]
    ... corps de la figure ...
\end{figure}
```

Pour un tableau, c'est le même principe, mais avec l'environnement table.

```
\begin{table}[option de placement]
... corps de la figure ...
\end{table}
```

# Les flottants

LATEX essaye d'optimiser le placement des flottants, ce qui ne donne pas forcément le résultat auquel on peut s'attendre...

Il existe des paramètres de placement qui donnent un plus grand degré de contrôle de placement des éléments flottants.

Paramètre de position	Permission	
h (here)	Position à l'endroit auquel le flottant apparaît dans le texte source.	
t (top)	Position en haut de la page	
b (bottom)	Position en bas de la page.	
p (page)	Place sur une page particulière réservée aux flottants	
!	Passe outre les paramètres internes que LATEX utilise pour détermi-	
	ner une position optimale des flottants	

Les flottants peuvent porter un titre grâce à \caption et se faire référence grâce à \label.

### Les images

Pour insérer une image, on utilise le package graphicx :

```
\usepackage{graphicx}
```

La figure est ajoutée en utilisant la commande : \includegraphics[option] {nom\_de\_1\_image}.

Diverses options de taille existent. Par exemple :

```
\includegraphics[scale=0.5]{image.jpg} % 50% de la taille initiale
    de l'image
\includegraphics[width=10cm]{image.jpg} % largeur de 10cm
\includegraphics[height=10cm]{image.jpg} % hauteur de 10cm
\includegraphics[angle=90]{image.jpg} % rotation d'un quart de tour
\includegraphics[width=0.8\textwidth]{image.jpg} % 80% de la
    largeur habituelle du texte.
```

### Les images

En général, on place les images dans un répertoire dédié par exemple « images ».

L'exemple suivant permet de créer une figure titrée, référencée et centrée :

```
\begin{figure}[htpb]
  \center
  \includegraphics[width=\textwidth]{./images/nomImage}
  \caption{\label{fig:nomCourt}Titre de la figure}
\end{figure}
```

tro Diagrammes LATEX Conclusion

### Les tableaux

L'environnement de base pour les tableaux en LaTeX est tabular. Il prend un argument obligatoire qui spécifie les colonnes du tableau. Cet argument est une suite de caractères qui peuvent être :

- un / pour une colonne alignée à gauche (left)
- un r pour une colonne alignée à droite (right)
- un c pour une colonne centrée (center)
- un  $p\{largeur\}$  pour une colonne avec un paragraphe en pavé (c'est-à-dire justifié, mais sans alinéa); la largeur est indiquée de manière classique, par exemple  $p\{3cm\}$
- un | si les colonnes doivent être séparées par un séparateur vertical

Chaque ligne du tableau est représentée par les valeurs des colonnes séparées par des esperluettes & et terminées par deux contre-obliques \\ (exprimant le saut de ligne). On peut aussi ajouter des séparateurs horizontaux avec la commande \hline.

Intro Diagrammes ATEX Conclusion

### Les tableaux : exemple

```
\begin{table}[htpb]
  \center
  \begin{tabular}{|1|1|p{5cm}|}
    \hline
    a & bbb & grande colonne\\
    \hline
    c & d & \\
    e & f & \\
    \hline
    \end{tabular}
\end{tabular}
\end{table}
```

а	bbb	grande colonne
С	d	
e	f	

D'autres environnements existent et sont potentiellement plus adéquats en fonction du type de tableau que vous voulez faire : array, tabularx, tabulary, etc.

# Les mathématiques

Au sein d'un texte, on peut mettre une expression mathématique entre des \$. Dans ce cas, celle-ci apparaît au sein du texte en cours et prend un format propre aux mathématiques.

Je dis que 
$$5+2=7$$
 ou  $$5+2=7$ \$.

Je dis que 
$$5+2=7$$
 ou  $5+2=7$ .

En mode mathématique, on peut utiliser des commandes permettant de faire des symboles usuels de mathématique. Par exemple, \sin, \cos, \log, \theta, \phi.

# Les mathématiques

Pour séparer une expression mathématique du texte en cours, il faut utiliser \$\$.

Le nombre  $\pi$  vaut environ 3,14. 4

Le nombre  $\pi$  vaut environ 3, 14.

$$a+b+c=d$$

Il existe un certain nombre d'environnements permettant de traiter des mathématiques, par exemple, l'environnement equation pour afficher une équation et l'environnement align qui permet d'afficher des équations de sorte que le "=" se trouve au même niveau sur chaque ligne. Pour utiliser l'environnement align, on doit ajouter le package amsmath.

# Ajouter une bibliographie

Une bibliographie est une liste de références qui identifie tous les documents nécessaires que vous avez utilisés pour la réalisation de votre travail (rapport, article, ...).

Il existe deux méthodes pour créer une bibliographie en LATEX :

- thebibliography peut être utilisée si on a peu de références
- BibTeX est utilisée dans la plupart des cas

# L'environnement thebibliography

```
\begin{document}

Je cite l'item \cite{norman}.

\begin{thebibliography}{1}
\bibitem{impj} The Japan Reader {\em Imperial Japan 1800-1945}
    1973: Random House, N.Y.
\bibitem{norman} E. H. Norman {\em Japan's emergence as a modern state} 1940: International Secretariat, Institute of Pacific Relations.
\end{thebibliography}
\end{document}
```

Je cite l'item [2].

#### Références

- [1] The Japan Reader Imperial Japan 1800-1945 1973: Random House, N.Y.
- [2] E. H. Norman Japan's emergence as a modern state 1940: International Secretariat. Institute of Pacific Relations.

tro Diagrammes LATEX Conclusion

### BibTeX

On crée un fichier .bib, qui contient des références décrites selon un standard relativement simple.

Il y a plusieurs types de référence :

- book : pour un livre
- inProceedings : pour un article d'une conférence
- article : pour un article d'un magazine ou d'une revue
- inbook : une partie d'un livre
- ...
- misc : pour les documents qui ne correspondent à aucune des autres catégories listées, notamment pour les sites web

Suivant le type, certains champs sont obligatoires (author, title, year, publisher pour book; author, title, booktitle, year pour inProceedings, etc.).

# Exemple de contenu d'un fichier .bib

```
@book{Mitchell97,
  author = {Mitchell, Tom M.},
  isbn = \{978 - 0 - 07 - 042807 - 2\}.
  publisher = {McGraw-Hill},
  title = {Machine Learning},
  year = 1997
@inproceedings {aue 07 dbpedia,
  author = {Auer, S. and Bizer, C. and Kobilarov, G. and Lehmann, J
      . and Cyganiak, R. and Ives, Z.},
  booktitle = {6th International Semantic Web Conference (ISWC)},
  pages = \{722 - 735\}.
  title = {{DBpedia: A Nucleus for a Web of Open Data}},
  year = 2008
Omisc{WinNT.
  title = {{MS Windows NT} Kernel Description},
  howpublished = {\url{http://web.archive.org/web
      /20080207010024/}}.
  note = {Accessed: 2010-09-30}
                                                                   50 / 80
```

### Citer les références

#### Dans le document :

- Pour citer une référence \cite{Nom\_de\_référence}
- Pour citer une page ou un chapitre\cite[p. le numéro de page] {Nom\_de\_référence}
- Pour citer plusieurs références en même temps :
   \cite{Nom\_de\_référence 1, Nom\_de\_référence 2}
- Pour citer que des références soient dans la bibliographie sans qu'on les ait citées : \nocite{Nom\_de\_référence}

#### À la fin du document :

```
\bibliographystyle{nom_du_style}
\bibliography{nom_du_fichier_bib}
```

Il existe différents style d'affichage de la bibliographie : voir exemples sur https://fr.overleaf.com/learn/latex/Bibtex\_bibliography\_styles.

# Compilation de BibTeX

Pour mettre à jour la bibliographie, il faut compiler avec :

- pdflatex nomDuFichierTex.tex
- bibtex nomDuFichierTex
- pdflatex nomDuFichierTex.tex
- pdflatex nomDuFichierTex.tex

Sur les éditeurs, il y a en général des touches spécifiques, par exemple sur l'éditeur TeXworks, il suffit simplement d'utiliser

« pdfLaTeX+MakeIndex+BibTeX ».

#### Démonstration

```
\documentclass[12pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage{hyperref}
\begin {document}
Je cite la référence \cite{WinNT} ainsi que les références \cite{
    aue07dbpedia,Mitchell97}.
\bibliographystyle{abbrv}
% \bibliographystyle {alpha}
\bibliography{biblio}
\end{document}
```

# Le package natbib

Le paquetage natbib (\usepackage{natbib}) permet d'utiliser d'autres commandes de citation.

```
Obook{goossens93,
author = "M. Goossens and F. Mittelbach and A. Samarin",
title = "The LaTeX Companion",
year = "1993",
publisher = "Addison-Wesley",
address = "Reading, Massachusetts"}
```

Commandes	Sortie
\citet{goossens93}	Goossens et al. (1993)
\citep{goossens93}	(Goossens et al., 1993)
\citet*{goossens93}	Goossens, Mittlebach, and Samarin (1993)
\citep*{goossens93}	(Goossens, Mittlebach, and Samarin, 1993)
\citeauthor{goossens93}	Goossens et al.
\citeauthor*{goossens93}	Goossens, Mittlebach, and Samarin
\citeyear{goossens93}	1993
\citeyearpar{goossens93}	(1993)
\citealt{goossens93}	Goossens et al. 1993

# Citer un algorithme

Il existe plusieurs paquetages permettant de citer un algorithme. Nous étudierons l'un d'entre eux : algorithm2e.

Il s'exploite via \usepackage{algorithm2e}.

#### Algorithme 1 : Trouver le maximum dans une séquence finie d'entiers

```
Input: A finite set A = \{a_1, a_2, \dots, a_n\} of integers Output: The largest element in the set

1 max \leftarrow a_1

2 for i \leftarrow 2 to n do

3 | if a_i > max then

4 | max \leftarrow a_i

5 | end

6 end
```

7 return max

```
\begin{algorithm}
\DontPrintSemicolon
\WinfA set C = \{c 1, c 2, \{dots, c r\}
     \}$ of denominations of coins.
     where ci > c > 1
     and a positive number $n$}
\KwOut{A list of coins $d 1.d 2.\ldots.d
     k, such that \sum_{i=1}^k d_i = n
     $ and $k$ is minimized}
$C \gets \emptyset$\;
\For {$ i \gets 1$ \textbf {to} $r$}{
 \While{$n \geq c_i$} {
    $C \gets C \cup \{c_i\}$\;
    $n \gets n - c i$\:
\Return{$C$}\:
\caption{Algorithme de rendu de monnaie}
\end{algorithm}
```

```
monnaie

Input: A set C = \{c_1, c_2, \dots, c_r\} of denominations of coins, where c_i > c_2 > \dots > c_r and a positive number n

Output: A list of coins d_1, d_2, \dots, d_k, such that \sum_{i=1}^k d_i = n and k is minimized

1 C \leftarrow \emptyset

2 for i \leftarrow 1 to r do

3 | while n > c_i do
```

 $C \leftarrow C \cup \{c_i\}$  $n \leftarrow n - c_i$ 

end

8 return C

Algorithme 2 : Algorithme de rendu de

```
\begin{algorithm}
\DontPrintSemicolon
\KwIn{A sequence of integers $\langle a
     _1, a_2, \ldots, a_n \rangle$}
\KwOut{The index of first location with
     the same value as in a previous
     location in the sequence}
$location \gets 0$\;
$i \gets 2$\:
\While{$i \leq n$ \textbf{and} $location
     = 0$}{
  $ j \gets 1$\;
  \While{$ j < i$ \textbf{and} $location =
    % The "u" before the "If" makes it so
          there is no "end" after the
         statement, so the else will
         then follow
    \uIf{a_i = a_j$}{
      $location \gets i$\;
    \Else{
      $ j \gets j + 1$\;
  $i \gets i + 1$\;
\Return{location}\:
\caption{Trouver le 1er élément dupliqué
     dans une séquence d'entiers}
\end{algorithm}
```

```
dupliqué dans une séquence d'entiers
    Input: A sequence of integers
             \langle a_1, a_2, \ldots, a_n \rangle
    Output: The index of first location
                with the same value as in a
                previous location in the
                sequence

    location ← 0

 2i \leftarrow 2
 3 while i \leq n and location = 0 do
         i \leftarrow 1
         while j < i and location = 0 do
               if a_i = a_i then
                    location \leftarrow i
               else
                   j \leftarrow j + 1
               end
10
11
         end
         i \leftarrow i + 1
12
13 end
14 return location
```

Algorithme 3 : Trouver le 1er élément

```
\begin{algorithm}
\DontPrintSemicolon
\KwIn{A sequence of integers $\langle a
     _1, a_2, \ldots, a_n \rangle$}
\KwOut{The index of first location with
     the same value as in a previous
     location in the sequence}
$location \gets 0$\;
$i \gets 2$\:
\While{$i \leg n \land location = 0$}{
  $ | \gets 1$\;
  \While{$ i < i \land location = 0$}{
    % The "l" before the If makes it so
          it does not expand to a second
          1. 1. 2. 0
    \label{lif} {a_i = a_j$}{
      $location \gets i$\;
    \lElse{
      $j \gets j + 1$\;
  $i \gets i + 1$\;
\Return{location}\:
\caption{Trouver le 1er élément dupliqué
     dans une séquence d'entiers}
\end{algorithm}
```

```
Algorithme 4 : Trouver le 1er élément
    dupliqué dans une séquence d'entiers
    Input: A sequence of integers
             \langle a_1, a_2, \ldots, a_n \rangle
    Output: The index of first location
                with the same value as in a
                previous location in the
                sequence
 1 location \leftarrow 0
 2i \leftarrow 2
    while i < n \land location = 0 do
         i \leftarrow 1
         while i < i \land location = 0 do
               if a_i = a_i then location \leftarrow i
               else i \leftarrow i + 1
         end
10
         i \leftarrow i + 1
11
12 end
13 return location
```

### Paquetage algorithm2e : options

Diverses options existent, par exemple:

```
\usepackage[linesnumbered, ruled, french, onelanguage]{algorithm2e}
```

- linesnumbered permet de numéroter les lignes
- ruled permet un affichage avec des lignes séparatrices
- french traduit permet d'avoir « algorithme » au lieu de « algorithm » par défaut
- onelanguage traduit les mots-clés<sup>1</sup>
- 1. Attention, le terme « output » n'est pas traduit dans la version française. Pour le traduire, il faut ajouter les 3 lignes ci-dessous après les usepackage :

```
\makeatletter
\g@addto@macro{\@algocf@init}{\SetKwInput{KwOut}{Sortie}}
\makeatother
```

### Citer son code

On peut utiliser le paquetage listings (\usepackage{listings}) pour citer du code.

#### Par exemple:

```
\begin{lstlisting}
public class MaClasse{
   private int a = 5;
}
\end{lstlisting}
```

#### donne

```
public class MaClasse{
   private int a = 5;
}
```

# Ajout d'options

On peut ajouter des options, notamment le langage de ce qui est cité.

### Par exemple:

```
\begin{lstlisting}[language=Java, numbers=right,
   numberstyle=\tiny, frame=single, linewidth=7cm]
public class MaClasse{
   private int a = 5;
\end{lstlisting}
```

3

#### donne

```
public class MaClasse{
   private int a = 5;
                                  2
```

### Citer un fichier

On peut aussi directement citer un fichier.

```
\lstinputlisting[language=Java,frame=single]{Hello.java}
```

# Faire une figure

Tikz est une librairie LATEX permettant de réaliser des figures vectorielles de manière simple.

Pour l'utiliser, on charge le paquetage :

```
\usepackage{tikz}
```

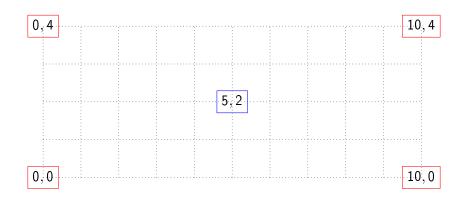
et on réalise les figures grâce à l'environnement tikz, qu'on place en général dans une figure :

```
\begin{figure}[htpb]
\begin{tikzpicture}
    % commandes
\end{tikzpicture}
\caption{Le titre}
\label{leLabel}
\end{figure}
```

Intro Diagrammes LATEX Conclusion

# Le paquetage Tikz

Chaque figure possède une grille (invisible), que l'on utilise pour positionner des objets avec des coordonnées (x,y) (non nécessairement entières).



# Le paquetage Tikz

Pour réaliser des figures simples, on utilise en particulier les commandes :

```
\node (identifiant 1) at (x,y) {texte};
\coordinate (identifiant 2) at (x',y');
\draw (identifiant 1) -- (identifiant 2);
```

#### qui ont pour effet, respectivement :

- d'ajouter à la figure un nœud contenant le texte donné, centré en position (x,y)
  - de définir un point (non représenté sur la figure) en position (x',y'), pour référence future
  - d'ajouter à la figure un lien entre ces deux entités.

Chaque commande doit être terminée par un point-virgule («; »).

Intro Diagrammes Conclusion

# Le paquetage Tikz

Les commandes peuvent prendre des options, notamment entre crochets ( $\ll$  [...]  $\gg$ ).

#### Par exemple:

```
\node[circle, draw=blue, color=red] (identifiant 1)
\draw[->] (identifiant 1) -- (identifiant 2);
```

#### permettant respectivement

- d'écrire le texte donné en rouge, entouré d'un cercle bleu
- d'ajouter une flèche à la fin du lien

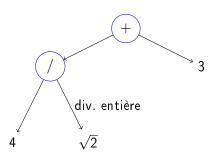
Plus d'informations sur http://math.et.info.free.fr/TikZ/bdd/TikZ-Impatient.pdf

### Exemple

```
\begin{tikzpicture}
  % Grille
  \draw[dotted] (0,0) grid (5,3);
  % Coins
  \node[rectangle, draw=red] (coin 1) at (0,0) {$0,0$};
  \node[rectangle, draw=red] (coin 2) at (0,3) {$0,3$};
  \node[rectangle, draw=red] (coin 3) at (5,0) {$5,0$};
  \node[rectangle, draw=red] (coin 4) at (5,3) {$5,3$};
  % Coin special
  \coordinate (debut fleche) at (-1.5.0);
  \draw[->, thick, blue] (debut fleche) -- (coin 1);
  % Noeuds (sauf coins)
  \node (racine) at (3.3) {$3.3$};
  \node (fils gauche) at (1,2) {$1,2$};
  \node (feuille 2) at (2,0) {$2,0$};
  \node (feuille 3) at (5.2) {$5.2$};
  % Aretes
  \draw[dashed] (racine) -- (fils gauche);
  \draw[dashed] (fils gauche) -- (coin 1);
  \draw[dashed] (fils gauche) -- (feuille 2);
  \draw[dashed] (racine) -- (feuille 3);
\end{tikzpicture}
                                                                                    5,0
```

### Exemple

```
\begin{tikzpicture}
    % Noeuds de l'arbre
    \node[circle, draw=blue] (racine) at (3,3) {$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{$\pmathscr{{\pmathscr{$\pmathscr{$\pmathscr{{\pmathscr{$\pmathscr{{\pmathscr{$\pmathscr{{\pmathscr{$\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmathscr{{\pmath}}}}}}
```



### Faire une présentation

Beamer est une classe de LATEX permettant de réaliser des présentations au format pdf. Elle propose de nombreux thèmes de présentation donnant une apparence soignée et agréable. Beamer possède une syntaxe spéciale pour définir les pages ( « diapositives »), appelées frames.

La compilation s'effectue comme pour un document LATEX standard.

### Beamer

```
\documentclass{beamer}
\usepackage[french]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usetheme{Warsaw}
\begin{document}
\begin{frame}
    Voici votre première
        page de présentation
        en LaTeX !
\end{frame}
\end{document}
```

Voici votre première page de présentation en LaTeX !

o Diagrammes LATEX Conclusion

### Les thèmes et couleurs

Il est possible de personnaliser complètement l'apparence de son diaporama mais il est recommandé pour débuter d'utiliser les thèmes fournis avec Beamer. Ceux-ci se divisent en quatre grandes catégories :

- Thème de présentation globale qui gère la totalité du diaporama : \usetheme{nom\_du\_thème}: Antibes, Berkeley, Berlin, Madrid, etc.
- Thème de couleur permettant de modifier les couleurs de base d'un thème global ou une partie seulement des couleurs selon les thèmes : \usecolortheme{nom\_de\_couleur} : beaver, dolphin, orchid, etc.
- Thème externe gère les en-têtes et pieds-de-page, titre de la page, etc.: \useoutertheme{nom\_de\_thème\_externe}: infolines, shadow, smoothbars, etc.
- Thème interne gère l'apparence des éléments tels que les listes, la table des matières, les notes, la bibliographie : \useinnertheme{nom\_de\_thème\_interne} : rectangles, circles, inmargin, rounded.

# Les titres de diapositives

Pour mettre un titre, on utilise la commande \frametitle.

```
\begin{frame}
  \frametitle{Titre de
      slide}
  Contenu du slide
  \end{frame}
```

```
Titre de slide
   Contenu du slide
```

### Les blocs

Beamer propose des environnements de bloc.

```
\begin{block}{Un bloc normal} % Bloc normal
A utiliser normalement, selon vos envies.
\end{block}
\begin{alertblock}{Bloc alerte} % Bloc alerte rouge
A utiliser pour alerter.
\end{alertblock}
\begin{exampleblock}{Un bloc exemple} % Bloc exemple vert
Suivi d'un exemple.
\end{exampleblock}
```

#### Un bloc normal

A utiliser normalement, selon vos envies.

### Bloc alerte

A utiliser pour alerter.

### Un bloc exemple

Suivi d'un exemple.

### La page de titre

La page de titre se construit comme dans les documents standard, grâce à des variables

```
\documentclass{beamer}
\title{Mon titre}
\author{Mon nom}
\institute{Unicaen}
\date{ma date}
\begin{document}
\frame{\titlepage}
\end{document}

madate
```

De même, le sommaire se construit à partir des sections et sous-sections dans le corps du document, avec la commande \tableofcontents.

# L'affichage différé : quelques exemples

La commande \pause permet de geler l'affichage dans la présentation, en attendant d'aller plus loin.

```
\begin{frame}
Voici ma première idée
\pause
Voici la deuxième idée
\pause
Voici la troisième idée
\end{frame}
```

La commande \onslide<...> est utilisée pour spécifier sur quelles diapositives le contenu spécifique apparaîtra.

```
\onslide<2> ce qui suit ne sera que sur la couche 2.
\onslide<2-4> ce qui suit sera de la couche 2 à 4.
\onslide<2,4> ce qui suit sera sur la couche 2 et 4.
\onslide<2-> ce qui suit sera de la couche 2 jusqu'à la fin de page.
\onslide<-3> ce qui suit sera du début de page jusqu'à la couche 3.
\onslide<2-3,5-7> ce qui suit sera sur les couches 2,3,5,6,7.
```

### Multi-colonnes

L'environnement minipage est pour cela très pratique.

```
\begin{minipage}[c]{0.48\textwidth}
  contenu de la colonne de gauche
\end{minipage}\hfill
\begin{minipage}[c]{0.48\textwidth}
  contenu de la colonne de droite
\end{minipage}
```

On peut également employer l'environnement columns, spécialement dédié aux multi-colonnes.

```
\begin{columns}
\begin{column}{6cm}
Contenu de ma première colonne
\end{column}
\begin{column}{6cm}
Contenu de ma deuxième colonne
\end{column}
\end{column}
\end{column}
```

### LaTeX: conclusion

Cette présentation introduit quelques éléments de LATEX.

On apprend par la pratique et avec le temps!

On peut exploiter bien plus de paquetages avec LATEX, c'est très bien documenté.

### Rapport:

- Prendre du recul sur son travail
- Important d'avoir des éléments visuels : figures, tableaux, diagrammes, algorithmes, etc.
- Les éléments visuels ne suffisent pas : il faut TOUJOURS du texte explicatif pour chaque image!

#### Diaporama:

- Aider le spectateur à comprendre
- Aider le présentateur à présenter
- Aider à la compréhension pour un spectateur absent

Pensez à numéroter les pages (rapport et diaporama)!



Diagrammes Structurels ou Diagrammes statiques.

https:

ntrps. //www.uml-sysml.org/diagrammes-uml-et-sysml/ diagramme-uml/structurel-ou-statique/. Accédé en : 07/2022.



Documentation algorithm2e.

https://distrib-coffee.ipsl.jussieu.fr/pub/ mirrors/ctan/macros/latex/contrib/algorithm2e/ doc/algorithm2e.pdf Accédé en: 07/2022.



Documentation Beamer.

http://tug.ctan.org/macros/latex/contrib/ beamer/doc/beameruserguide.pdf. Accédé en: 07/2022.



Documentation BibTeX. http://www.bibtex.org/.

Accédé en : 07/2022.



Documentation LaTeX.

https://www.latex-project.org/help/documentation/usrguide.pdf. Accédé en : 07/2022.



Documentation listings.

https://ctan.tetaneutral.net/macros/latex/contrib/listings/listings.pdf.
Accédé en: 07/2022.



Documentation TikZ.

https://texample.net/media/pgf/builds/ pgfmanualCVS2012-11-04.pdf. Accédé en: 07/2022.



Tik Z pour l'impatient.

http://math.et.info.free.fr/TikZ/bdd/ TikZ-Impatient.pdf. Accédé en: 07/2022.



UML: Langage de modélisation objet unifié. Cours n 3: Diagramme des classes et Cours n 10: Diagramme des paquetages. http://remy-manu.no-ip.biz/. Accédé en: 07/2022.