

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

Eric Thiémard

**heig-vd**  
Haute Ecole d'Ingénierie et de Gestion  
du Canton de Vaud

**Hes-so**  
Haute Ecole Spécialisée  
de Suisse occidentale

## L<sup>A</sup>T<sub>E</sub>X ?!?... heu, mais c'est quoi ?

Commençons par quelques illustrations :

- Exemples de documents, entièrement réalisés à l'aide de L<sup>A</sup>T<sub>E</sub>X, sur la page web du cours :

`http://sachem.heig-vd.ch/latex/`

Bien entendu, cette présentation a également été réalisée avec L<sup>A</sup>T<sub>E</sub>X.

- Environnement L<sup>A</sup>T<sub>E</sub>X que nous installerons plus tard.

## Organisation du cours

Dix leçons de deux périodes avec à chaque fois :

- Présentation de nouveaux éléments théoriques
- Mise en pratique sur vos machines

Thèmes abordés :

- Leçon 1 : introduction et installation
- Leçons 2 et 3 : bases et sélection de commandes fondamentales
- Leçons 4 à 6 : tableaux, figures et formules mathématiques
- Leçons 7 et 8 : graphiques et dessins
- Leçon 9 : réalisation de présentations
- Leçon 10 : extensions et sélection d'éléments avancés

## *Partie 1*

### *Introduction et installation*

## Un peu d'histoire...

C'est en 1977 que **Donald E. Knuth**, (déjà) célèbre professeur d'informatique théorique à l'université de Stanford (en Californie) (<http://www-cs-staff.stanford.edu/~knuth/>) reçoit par son éditeur les épreuves en vue de la réédition du second volume de son livre "The Art of Computer Programming".



Scandalisé par la qualité du résultat, il réalise que la typographie peut être vue comme une affaire d'informaticien et il décide de développer un système permettant à chacun de créer des documents scientifiques de qualité irréprochable.

Il se lance donc dans la création de  $\text{T}_{\text{E}}\text{X}$  et de METAFONT.

Initialement, il pensait s'en charger durant son semestre sabbatique de 1978...  
il y consacrera finalement environ 10 ans.

## De $\text{T}_{\text{E}}\text{X}$ à $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

$\text{T}_{\text{E}}\text{X}$  est un outil très puissant, mais assez rébarbatif pour l'utilisateur moyen...

Après quelques épisodes intermédiaires, c'est en 1985 que **L. Lamport** crée  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ , une extension de  $\text{T}_{\text{E}}\text{X}$ . En bref, il s'agit d'une collection de **macros** offrant à moindre effort l'accès à une grande partie de la puissance de  $\text{T}_{\text{E}}\text{X}$ . Nous utilisons aujourd'hui la **version** stabilisée  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$ .

Plusieurs projets d'extension sont en cours de réalisation, notamment

- $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}_3$  (<http://www.latex-project.org/>)
- $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$  (<http://tug.org/xetex/>)

Aujourd'hui, la quasi-totalité des livres et articles mathématiques sont écrits en  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  et plus généralement, dans le monde scientifique, cet heureux phénomène s'étend progressivement, si bien que  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  s'avère déjà être LE standard pour les publications en ingénierie, en informatique et en physique.

## Points positifs et négatifs

### Avantages

- Qualité professionnelle, multi-supports, multi-langues
- Encombrement minimal, portabilité, stabilité
- Libre de droit, gratuit, extensible
- Séparation contenu/forme, structuration automatique, maîtrisabilité

### Inconvénients

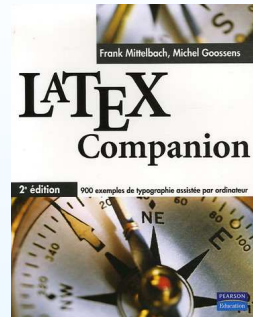
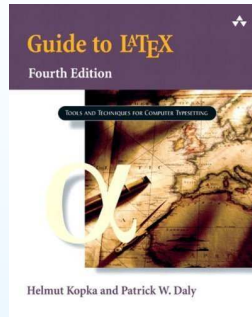
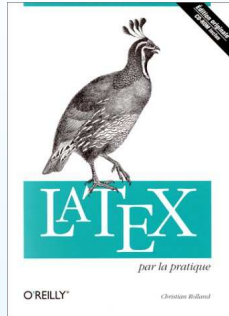
- Pas WYSIWYG (What You See Is What You Get)
- Partage de fichiers avec des non-convertis
- Premiers pas difficiles

## Documentation gratuite

On trouve tellement d'information sur  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  sur internet que nous pourrions presque nous limiter aux ressources **en français** :

- Deux documents de référence :
  - Une courte introduction à  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$  de Tobias Oetiker  
<http://www.ctan.org/tex-archive/info/lshort/french/>
  - Savoir  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  de Vincent Lozano  
<http://latex.developpez.com/cours/savoir-latex-sans-osser-demander/>
- Quelques FAQ pour les questions courantes :
  - <http://www.grappa.univ-lille3.fr/FAQ-LaTeX/>
  - <http://latex.developpez.com/faq/>
  - <http://www.cours.polymtl.ca/latex/latexfaq.html>
- Il existe également divers groupes d'utilisateurs, comme par exemple  
<http://www.gutenberg.eu.org/>

## Trois bon livres pour Noël



- L'AT&E par la pratique, C. Rolland, O'Reilly, Paris, 2<sup>e</sup> édition, 2000  
... en français, avec CD d'installation inclus
- Guide to L'AT&E, H. Kopka and P. W. Daly, Addison-Wesley, 4rd ed., 2003
- L'AT&E companion, F. Mittelbach et M. Goossens, Pearson, 2<sup>e</sup> édition, 2006  
... en français, avec CD d'installation inclus

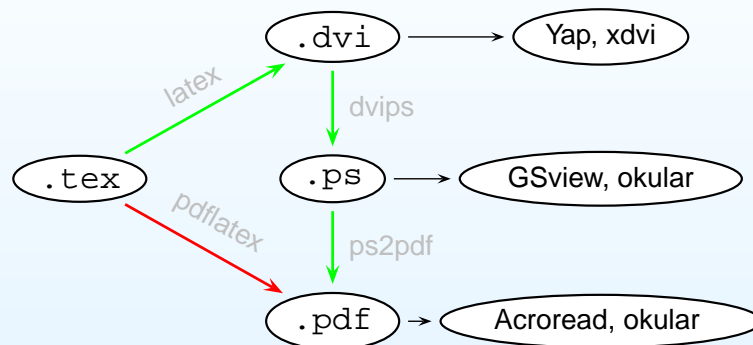
## Se procurer un environnement L'AT&E gratuitement

Il y a plusieurs possibilités, mais voici une solution pour les trois OS courants :

- Sous Linux, installer les packages suivants de votre distribution :
  - TeX Live (système L'AT&E)
  - Texmaker ou Kile (environnement d'édition/compilation)
  - Okular (visualisation des fichiers PDF et PostScript)
- Sous Windows, installer les logiciels suivants :
  - Acrobat Reader (visualisation des fichiers PDF)
  - Ghostscript (traitement des fichiers PDF et PostScript)
  - GSview (visualisation des fichiers PostScript)
  - MiKTeX (système L'AT&E)
  - Texmaker ou TeXnicCenter (environnement d'édition/compilation)
- Sous MacOSX, installer MacTeX et Texmaker

## Programmes et formats principaux

Source L'AT&E      Fichiers compilés      Visualisation / impression



Des environnements tels que Texmaker facilitent ces processus.

## Installation

Allons-y !

<http://sachem.heig-vd.ch/latex/>

## Partie 2

### Quelques éléments fondamentaux

## Structure d'un document $\text{\LaTeX}$

Tout document  $\text{\LaTeX}$  présente la **structure** suivante :

```
\documentclass[options]{classe du document}
    En-tête du document

\begin{document}
    Corps du document
\end{document}
```

Les **options** (facultatives, à séparer par des virgules) les plus courantes sont :

- 10pt, 11pt, 12pt                      Taille de base pour les caractères
- a4paper, a5paper, letterpaper, ...    Format du papier
- landscape                              Orientation “paysage” du papier
- oneside, twoside                      Document recto ou recto-verso

## Les classes de document

Il existe cinq **classes** standard de documents

- article
- report
- book
- letter
- slides

ainsi que de nombreuses autres élaborées par divers utilisateurs.

Chaque classe associe au document un certain **style** (paramétrable) et introduit un certain nombre de **commandes** spécifiques.

Par exemple, le présent document utilise la classe `prospect` paramétrée avec les couleurs, les polices de caractères, le logo,... que vous voyez.

## L'en-tête du document

L'**en-tête** du document est une zone dans laquelle on peut notamment

- spécifier certains **paramètres** comme la largeur des marges, le contenu d'un éventuel pied de page, les polices de caractères, si les pages doivent être numérotées en chiffres arabes ou romains, etc. ; on a par exemple

```
\title{Introduction à \LaTeX}           % On définit le titre du document
\setlength{\textheight}{24cm}          % Le texte sera de 24cm de hauteur
\setlength{\parskip}{1cm plus 0.2cm minus 0.2cm} % L'espace
                                              vertical avant un paragraphe est de 1cm ± 0.2 cm
```

- inclure des **packages** ajoutant à  $\text{\LaTeX}$  diverses fonctionnalités (écriture d'algorithmes, graphisme, partitions musicales, schémas électriques,...)

```
\usepackage[options]{nom du package}
```

- définir de nouvelles **commandes** ou redéfinir des commandes existantes

```
\renewcommand{\baselinestretch}{1.3} % Augmente l'interligne de 30%
```

## L'en-tête par l'exemple...

```
\documentclass[a4paper,12pt,landscape]{article}

\usepackage[utf8]{inputenc} % Encodage utf-8
\usepackage[T1]{fontenc}    % Nouvelle norme pour codage des caractères
\usepackage{lmodern}        % Nouvelle forme de la fonte Computer Modern
\usepackage[frenchb]{babel} % Règles typographiques françaises, césures

\title{\Huge Initiation à \LaTeX{}} % Définition du titre du document,
\author{\Large Eric Thiémard}      % du nom de l'auteur et de la date
\date{\today}                     % (ici la date de compilation...)

\pagestyle{plain} % Style par défaut : pied de page avec numéro de page

\begin{document} % Début du corps du document
\maketitle        % Affiche la page de titre selon définitions
\thispagestyle{empty} % La page de titre ne doit pas être ``décorée''
\newpage          % Passer à la page suivante
Hello world!      % Le texte du document
\end{document}    % Fin du corps du document
```

## Après compilation, on obtient la première page

Initiation à  $\text{\LaTeX}$   
Eric Thiémard  
2 septembre 2013

## ... et la seconde page

Hello world!

## Structuration des documents

Un document de la classe `article`, `report` ou `book` se décompose en **parties**, elles-mêmes divisées en **chapitres** (sauf pour `article`) qui se subdivisent eux-mêmes en **sections**, **sous-sections** et **sous-sous-sections**. L'aspect par défaut (reconfigurable) du résultat dépend de la classe choisie.

Le nommage et la numérotation hiérarchique de ces subdivisions s'effectue à l'aide de `\part{nom}`, `\chapter{nom}`, `\section{nom}`, `\subsection{nom}` et `\subsubsection{nom}`. Exemple :

## 2 Introduction

blabla

```
\section{Introduction}
blabla
\subsection{Remarques}
```

### 2.1 Remarques

## Références

Des **points de référence** (invisibles) peuvent être placés n'importe où dans le document à l'aide de la commande `\label{nom_du_label}`.

Il est possible de faire référence à la **page** à laquelle ce point a été placé avec `\pageref{nom_du_label}` que...

De même, pour faire référence à la **(sous)-section** dans laquelle il se trouve : `\ref{nom_du_label}`, on a...

Notons par avance que les commandes `\label{nom_du_label}` et `\ref{nom_du_label}` pourront également être utilisées pour faire référence à un numéro de figure, d'équation, de table, de théorème, de remarque, etc. Il suffit de placer le point de référence dans l'environnement en question (dans la légende de l'objet dans le cas d'une figure ou d'une table).

## Espacement

Les commandes `\vspace{distance}` et `\hspace{distance}` permettent d'insérer des **espaces** verticaux ou horizontaux (par exemple `\hspace{2.3cm}`), où la *distance* peut être notamment donnée en :

cm	centimètres (1 pouce = 2.54 cm)	in	pouces (1 in = 72.27 pt)
mm	millimètres (1 mm = 2.845 pt)	pt	points (1 pt = 0.351 mm)

Parmi les macros d'espacement **horizontaux**, on utilise souvent

<code>\,</code>	Le sixième d'un quad
<code>\quad</code>	La largeur d'un M dans la police courante
<code>\qquad</code>	Le double d'un quad

Notons également les commandes `\smallskip`, `\medskip` et `\bigskip` qui permettent respectivement l'insertion d'un petit, moyen ou gros espace **vertical** (dépendants du style utilisé, mais reparamétrables).

## Quelques commandes courantes

La commande `\maketitle` permet d'afficher une **page de titre** (dont l'apparence dépend de la classe du document) dont les éléments auront préalablement été donnés à l'aide de `\title{titre du document}`, `\author{nom de l'auteur}` et `\date{date à afficher}`.

La commande `\newpage` permet de passer à la **page suivante**.

La commande `\\` force un **retour à la ligne**. Elle ne peut pas être utilisée deux fois de suite (*i.e.* s'il n'y a pas de texte entre les deux utilisations).

Une ligne blanche dans le source provoque un changement de **paragraphe**. Le texte qui suit est donc légèrement décalé sur la droite (pour annuler ce **décalage**, insérer la commande `\noindent` au début du paragraphe).

Notons qu'en  $\text{\LaTeX}$ , les **espaces multiples** sont remplacés par un seul.

## Caractères spéciaux

Ce qui suit le caractère `%` sur une ligne du code source est un **commentaire**. Pour afficher le caractère `%` dans le document, on utilise la commande `\%`.

De manière similaire, d'autres caractères ont une **fonction spéciale** en  $\text{\LaTeX}$ , mais sont affichables par des commandes particulières. Il s'agit notamment de

Caractère	Commande	Caractère	Commande
\$	<code>\\$</code>	{	<code>\{</code>
&	<code>\&amp;</code>	}	<code>\}</code>
#	<code>\#</code>	_	<code>\_</code>
\	<code>\textbackslash</code>	œ	<code>\oe</code>
~	<code>\textasciitilde</code>	^	<code>\textasciicircum</code>

Remarque : divers packages, commandes ou familles de fontes permettent à  $\text{\LaTeX}$  d'être utilisé dans la plupart des **langues**.

Ångström (1814-1874)      `\AA`    `\ngstr`    `\o`    m (1814-1874)

## Tailles de caractères

Par défaut, dix **tailles de caractères** sont disponibles et les changements s'effectuent à l'aide des commandes suivantes :

<code>\tiny</code>	Eat me
<code>\scriptsize</code>	Eat me
<code>\footnotesize</code>	Eat me
<code>\small</code>	Eat me
<code>\normalsize</code>	Eat me (taille par défaut)
<code>\large</code>	Eat me
<code>\Large</code>	Eat me
<code>\LARGE</code>	Eat me
<code>\huge</code>	Eat me (comme <code>\LARGE</code> en prosper)
<code>\Huge</code>	Eat me (comme <code>\LARGE</code> en prosper)

## Tailles de caractères II

Au besoin, utiliser des accolades **pour** placer les changements de taille.

Au besoin, utiliser  
`{\scriptsize des accolades`  
`{\Large pour} placer}` les  
changements de taille.

Pour d'autres formes/tailles, c'est possible, mais souvent plus technique...

OK

```
\fontfamily{phv}  
\fontseries{bx}  
\fontsize{80}{0}  
\selectfont OK
```

Pour les **exposants** textuels, utiliser `\textsuperscript{}` :

Louis 1<sup>er</sup>

Louis 1`\textsuperscript{er}`

## Styles et polices de caractères

Il existe de nombreuses commandes pour changer de **style de caractères**, mais les plus courantes sont sans doute les suivantes :

*Texte en italique*

**Texte en gras**

Style typewriter

PETITES CAPITALS

Soulignement

```
\textit{Texte en italique}\\  
\textbf{Texte en gras}\\  
\texttt{Style typewriter}\\  
\textsc{Petites capitales}\\  
\underline{Soulignement}
```

Par défaut, la **famille de fontes** d'un document L<sup>A</sup>T<sub>E</sub>X est *Computer Modern* (ce n'est pas le cas de ces transparents). Il est bien sûr possible d'en changer, mais nous nous bornerons à signaler le fait que placer une commande telle que

```
\usepackage{times} \usepackage{newcent}  
\usepackage{helvet} \usepackage{palatino}
```

dans l'en-tête, permet de changer la famille principale de fontes du document.

## Partie 3

### Sélection de commandes classiques



## Trois environnements d'alignement

### Centré :

Les  
cours reprennent tout  
prochainement.

```
\begin{center}
  Les\\cours reprennent
  tout prochainement.
\end{center}
```

### Calé à gauche :

Les  
cours reprennent tout  
prochainement.

```
\begin{flushleft}
  Les\\cours reprennent
  tout prochainement.
\end{flushleft}
```

### Calé à droite :

Les  
cours reprennent tout  
prochainement.

```
\begin{flushright}
  Les\\cours reprennent
  tout prochainement.
\end{flushright}
```

## verbatim

L'environnement `verbatim` permet **d'interrompre l'interprétation** L<sup>A</sup>T<sub>E</sub>X et donc d'insérer tels quels (en style typewriter, en préservant les espaces et les sauts à la ligne) des listings tels que du code C ou L<sup>A</sup>T<sub>E</sub>X.

```
#include <stdlib.h>
```

```
int main() {
  printf("Hello\n");
}
```

```
\begin{verbatim}
#include <stdlib.h>
```

```
int main() {
  printf("Hello\n");
}
\end{verbatim}
```

Pour des chaînes courtes, on peut également utiliser `\verb` et placer la chaîne à afficher entre deux symboles identiques, deux `+` par exemple :

Commande `\LaTeX{ }`

Commande `\verb+\LaTeX{ }+`

## Compteurs

Il s'agit de **variables entières** utilisées pour représenter diverses grandeurs : page (numéro de la page courante), section (numéro de la section courante), etc. Ces grandeurs sont gérées automatiquement, mais il est possible de les **modifier** avec `\setcounter`, `\addtocounter` et `\stepcounter` ou d'**afficher** leur valeur sous différents formats :

### 6 Saut section

### 21 Saut de 14

### 23 Saut de 1

23 = xxiii = XXIII = w = W

```
\setcounter{section}{5}      % =5
\section{Saut section}      % +1
\addtocounter{section}{14}  % +14
\section{Saut de 14}        % +1
\stepcounter{section}       % +1
\section{Saut de 1}         % +1
\arabic{section} % = \thesection
= \roman{section} = \Roman{section}
= \alph{section} = \Alph{section}
```

## Numéros de page et notes de bas de page

La commande `\pagenumbering{arabic}` permet de **numéroter les pages** en chiffres arabes (à partir de 1). En remplaçant `arabic` par `roman` ou `alph` (respectivement Roman ou Alph), on obtient des chiffres romains ou des lettres en minuscules (respectivement en majuscules). Le compteur de page (modifiable) s'appelle `page`.

La commande `\footnote{texte}` permet d'insérer un texte en **note de bas de page** à partir d'un point de référence dans la page.

Fear is the path to the  
dark side<sup>a</sup> (Yoda).

<sup>a</sup>of the force... of course

```
Fear is the path to the dark
side\footnote{of the force...
of course} (Yoda).
```

Par défaut, la numérotation de ces notes est automatique, mais aisément modifiable en agissant sur le compteur `footnote`.



## Écrire de nouveaux théorèmes !

Dans un document, il peut être judicieux de structurer les choses en mettant en évidence des remarques, des observations, etc. La commande

`\newtheorem{Mon_Environnement}{Mon_Titre}` permet de définir un nouvel environnement `Mon_Environnement` qui, lorsqu'on l'utilise affiche en gras `Mon_Titre` et le compteur incrémenté :

**Remarque 1** Voir remarque 2.

**Remarque 2** Un package tel que `ntheorem` permet d'adapter l'apparence d'un tel environnement.

**Remarque 119** Initialisation du compteur via son nom

```
\newtheorem{rem}{Remarque}
\begin{rem}
  Voir remarque \ref{RemLook}.
\end{rem}
\begin{rem}\label{RemLook}
  Un package tel que \texttt{ntheorem} permet
  d'adapter l'apparence d'un tel environnement.
\end{rem}
\setcounter{rem}{118} % Nom du compteur = rem
\begin{rem}
  Initialisation du compteur via son nom
\end{rem}
```

## Table des matières

La commande `\tableofcontents` place la **table des matières** du document à l'endroit où elle est invoquée.

Par exemple, dans notre cas particulier, on obtient

## Table des matières

<b>2</b>	<b>Introduction</b>	<b>20</b>
2.1	Remarques . . . . .	20
<b>6</b>	<b>Saut section</b>	<b>31</b>
<b>21</b>	<b>Saut de 14</b>	<b>31</b>
<b>23</b>	<b>Saut de 1</b>	<b>31</b>

`\tableofcontents`

## Table des matières II

Suivant s'il s'agit d'un livre ou d'un court article, le **niveau de détail** que l'on souhaite afficher dans la table des matières n'est pas forcément le même... Faut-il aller jusqu'aux sections, aux sous-sections, aux sous-sous-sections ? Le compteur `tocdepth` (par défaut à 3) gère cette profondeur et peut être ajusté à l'aide de `\setcounter{tocdepth}{valeur}` pour augmenter/réduire le niveau de détail souhaité.

Sans entrer dans une présentation détaillée, signalons le fait que pour ajouter manuellement des éléments supplémentaires dans la table des matières, un moyen consiste à utiliser les commandes `\addcontentsline{ }{ }{ }` et/ou `\addtocontents{ }{ }`.

## Les listes simples

L'environnement `itemize` et `\item` permettent de construire des **listes** :

- En premier lieu, on note qu'il n'est pas possible de...
- Par contre,...

```
\begin{itemize}
\item En premier lieu, on note
  qu'il n'est pas possible de...
\item Par contre,...
\end{itemize}
```

Il est bien entendu possible de mettre une liste dans une liste :

- Disque vinyle
  - 33 tours
  - 45 tours
- CD

```
\begin{itemize}
\item Disque vinyle
  \begin{itemize}
    \item 33 tours
    \item 45 tours
  \end{itemize}
\item CD
\end{itemize}
```

## Listes numérotées et énumérations

De la même manière qu'`itemize`, les environnements `enumerate` et `description` permettent de construire des **listes numérotées** et des **énumérations**

- En premier lieu, on note qu'il...
  1. toto
  2. tutu

**label** dans un environnement description, on discute souvent les

**symbole** termes entre crochets

  3. titi
- Par contre,...
- En revanche,...

```
\begin{itemize}
\item En premier lieu, on note qu'il...
  \begin{enumerate}
  \item toto
  \item tutu
  \begin{description}
  \item[label] dans un environnement %
  description, on discute souvent les %
  \item[symbole] termes entre crochets %
  \end{description}
  \item titi
  \end{enumerate}
\item Par contre,...
\item En revanche,...
\end{itemize}
```

## Modification des compteurs d'énumération

Jusqu'à quatre niveaux d'imbrication sont envisageables. Les compteurs correspondants sont `enumi`, `enumii`, `enumiii` et `enumiv`.

100. Le compteur de liste `enumi` vaut 99, ce qui donne 100 au nouvel item.
101. Au nouvel item, il vaut 101
- (a) Le compteur de sous-liste `enumii` part de 1 = a
  - (b) Au nouvel item, il vaut 2
  - (m) On augmente `enumii` de 10. On ajoute 1 et il vaut 13
103. Ajoutons 1 et l'on obtient 103 au nouvel item

```
\begin{enumerate}
\setcounter{enumi}{99}
\item Le compteur de liste enumi vaut 99,
      ce qui donne 100 au nouvel item.
\item Au nouvel item, il vaut \theenumi
  \begin{enumerate}
  \item Le compteur de sous-liste enumii part
        de \arabic{enumii} = \theenumii
  \item Au nouvel item, il vaut 2
  \addtocounter{enumii}{10}
  \item On augmente enumii de 10. On
        ajoute 1 et il vaut \arabic{enumii}
  \end{enumerate}
\stepcounter{enumi}
\item Ajoutons 1 et l'on obtient \theenumi{}
      au nouvel item \end{enumerate}
```

## Les longueurs

Les longueurs sont des variables, généralement destinées à représenter la taille de quelque chose : `\textheight` (hauteur de la zone principale d'écriture de la page), `\parskip` (l'espace vertical ajouté avant un nouveau paragraphe), etc.

Contrairement aux compteurs, le nom d'une longueur commence par un `\` et un nom d'unité (cm, pt, etc., voir page 23) doit être spécifié pour ajuster une longueur à l'aide de la commande `\setlength` :

```
\setlength{\textheight}{23.8cm}
```

Il est possible d'affecter à une longueur un multiple d'une autre longueur :

```
\setlength{\textheight}{1.5\textrwidth}
```

Il est également possible d'ajouter une longueur (éventuellement négative) à une autre longueur :

```
\addtolength{\textheight}{10pt}
```

## Structure de la page

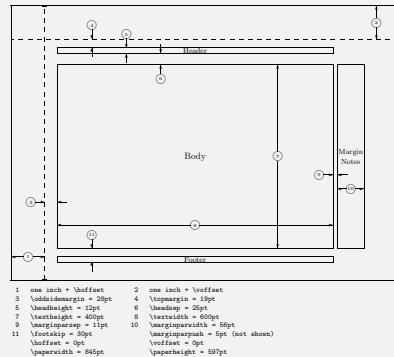
Après avoir ajouté `\usepackage{layout}` dans l'en-tête, la commande `\layout` permet d'afficher (là où elle est invoquée) la maquette de la page : dimensions et placement de la zone principale, des marges, de l'en-tête et du pied de page. Par exemple, en compilant le document

```
\documentclass[a4paper,landscape]{article}
\usepackage{layout}
```

```
\setlength{\oddsidemargin}{1cm}
\setlength{\textrwidth}{600pt}
\setlength{\textheight}{400pt}
\setlength{\marginparwidth}{2cm}
```

```
\begin{document}
\layout
\end{document}
```

... on obtient la maquette, qu'il suffit d'ajuster...



## Multicolonnage

Ajouter `\usepackage{multicol}` dans l'en-tête du document, puis

Comme pour la plupart des commandes LaTeX, multicol accepte diverses options pour personnaliser son œuvre :

espacement entre les colonnes, placement d'une ligne de largeur arbitraire entre les colonnes,...

```
\begin{multicols}{2}
Comme pour la plupart des
commandes LaTeX, multicol
accepte diverses options pour
personnaliser son {\oe}uvre~:
espacement entre les colonnes,
placement d'une ligne de largeur
arbitraire entre les colonnes,...
\end{multicols}
```

Second exemple avec ajustement de l'espace entre les colonnes et de la largeur des lignes.

```
\setlength{\columnsep}{0.5cm}
\setlength{\columnseprule}{0.6pt}
\begin{multicols}{3}
Second exemple avec ajustement
de l'espace entre les colonnes
et de la largeur des lignes.
\end{multicols}
```

## Partie 4

### Tableaux et figures

## Les colonnes

Le moyen le plus courant pour créer un **tableau** est d'utiliser l'environnement

```
\begin{tabular}{format_colonnes}
lignes du tableau
\end{tabular}
```

où `format_colonnes` est un argument obligatoire indiquant le **formatage** des colonnes. Les principaux éléments pouvant intervenir sont

- "`l`" pour créer une colonne dont le contenu doit être justifié à **gauche**,
- "`r`" pour créer une colonne dont le contenu doit être justifié à **droite**,
- "`c`" pour créer une colonne dont le contenu doit être **centré**,
- "`|`" pour tracer une **ligne verticale** entre deux colonnes,
- "`*{nb}{cols}`" est juste un moyen d'**abrégier l'écriture**.  
Par exemple `*{7}{|c|}` est équivalent à `|c|c|c|c|c|c|c|`

## Les lignes

Au niveau des lignes, on utilisera

- **&** pour **séparer** les éléments d'une ligne,
- **\\** pour passer à la **ligne suivante**,
- **\hline** pour insérer une **ligne horizontale** entre deux lignes,
- **\vline** pour insérer une **ligne verticale** de même hauteur que la cellule courante à l'endroit exact où la commande est placée.

Prix	Lu	Ma	Me
Or	173	65	18
Fer	2	9	7
x y		A	z

```
\begin{tabular}{|c|rr||l|}
\hline
Prix & Lu & Ma & Me\\
Or & 173 & 65 & 18\\
Fer & 2 & 9 & 7\\
x y & \vline & A & z\\
\end{tabular}
```

## Positionnement

En fait, `tabular` admet un paramètre optionnel `position`

`\begin{tabular}[position]{format_colonnes}`

qui peut prendre les valeurs suivantes :

- `[t]` pour un aligner le **haut** du tableau avec la ligne de texte courante,
- `[b]` pour un aligner le **bas** du tableau avec la ligne de texte courante,
- *rien* pour **centrer** le tableau par rapport à la ligne de texte courante.

Au centre 

a	b
c	d

 en haut 

a	b
c	d

 en bas 

a	b
c	d

 xyz

Au centre `\begin{tabular}{c|c}`  
`a & b\\ \hline c & d`  
`\end{tabular}`  
 en haut `\begin{tabular}[t]{c|c}`  
`a & b\\ \hline c & d`  
`\end{tabular}`  
 en bas `\begin{tabular}[b]{c|c}`  
`a & b\\ \hline c & d`  
`\end{tabular}` xyz

## Paragraphes

Il est souvent nécessaire de pouvoir placer des **paragraphes de texte** dans un tableau. Avec `tabular`, il suffit de créer une colonne de la forme

`p{largeur}`

Abc	850 ms	Le temps de latence est beaucoup trop important pour pouvoir garantir les performances requises.
Xyz	260 ms	C'est mieux, mais encore relativement important.

```
\begin{tabular}{|c|l|p{3.9cm}|}
\hline
Abc & 850 ms & Le temps de latence est beaucoup trop important pour pouvoir garantir les performances requises.\\
Xyz & 260 ms & C'est mieux, mais encore relativement important.\\
\hline
\end{tabular}
```

## Lignes horizontales partielles

Nous connaissons déjà `\hline` pour tracer une ligne horizontale sur toute la largeur d'un tableau, mais il existe également une commande

`\cline{x-y}`

qui permet de **relier** par une **ligne horizontale** les **colonnes** `x` à `y`.

Prix	Lu	Ma	Me	Je
Or	17	16	17	19
Fer	12	9	7	12
Eau	19	21	25	27

```
\begin{tabular}{|c|rrrr|}
\hline
Prix & Lu & Ma & Me & Je\\
\cline{2-4}
Or & 17 & 16 & 17 & 19\\
\cline{1-2}\cline{4-5}
Fer & 12 & 9 & 7 & 12\\
Eau & 19 & 21 & 25 & 27\\
\hline
\end{tabular}
```

## Fusion horizontale de cellules

`\multicolumn{nb_col}{format_colonne}{contenu}`  
permet de **fusionner** `nb_col` **colonnes** en une seule, de la formater selon `format_colonne` et d'y placer un certain contenu.

	Prix	
	HT	TTC
Or	67.-	75.-
Fer	9.-	11.-

```
\begin{tabular}{|l|r|r|}
\hline
& \multicolumn{2}{c|}{Prix} \\
\cline{2-3}
& HT & TTC \\
Or & 67.- & 75.- \\
Fer & 9.- & 11.- \\
\end{tabular}
```

Remarque : de manière similaire, le package et l'environnement `multirow` permettent la fusion verticale de cellules.

## Coloration des lignes

Ajouter dans l'en-tête du document `\usepackage{colortbl}` permet de **colorer ou d'ombrer des lignes** d'un tableau respectivement avec

`\rowcolor{couleur}`  
`\rowcolor[gray]{intensité}`

Jour	Lu	Ma
Min	5	4
Max	9	6

```
\begin{tabular}{|l|rr|}\hline
\rowcolor[gray]{0.5} Jour & Lu & Ma \\
\hline
\rowcolor[gray]{0.6} Min & 5 & 4 \\
\hline
\rowcolor{yellow} Max & 9 & 6 \\
\hline
\end{tabular}
```

## Coloration des colonnes

L'inclusion du package `\usepackage{colortbl}` permet également de **colorer ou d'ombrer des colonnes** d'un tableau respectivement avec

`>\columncolor{couleur}`  
`>\columncolor[gray]{intensité}`

Jour	Lu	Ma
Min	5	4
Max	9	6

```
\begin{tabular}{|l|rr|}
>\columncolor{yellow}1
>\columncolor[gray]{0.7}r
>\columncolor[gray]{0.5}r
Jour & Lu & Ma \\
Min & 5 & 4 \\
Max & 9 & 6 \\
\end{tabular}
```

## Diagonales

Ajouter dans l'en-tête du document `\usepackage{slashbox}` afin d'utiliser les commandes suivantes qui permettent de **séparer** une cellule en **deux parties** séparées par une diagonale (il est alors possible d'utiliser la première cellule pour donner la signification des lignes et des colonnes) :

`\slashbox{bas}{haut}`  
`\backslashbox{bas}{haut}`

Carats \slashbox{bas}{haut}	Prix	HT	TTC
	18	60	85
	24	90	120

```
\begin{tabular}{|c|rr|}
\hline
\backslashbox{Carats}{Prix}
& HT & TTC \\
18 & 60 & 85 \\
24 & 90 & 120 \\
\hline
\end{tabular}
```

## Rotations

Il arrive que la page soit trop étroite pour pouvoir placer la légende des colonnes d'un tableau. Une solution consiste à faire **tourner le contenu** des cellules concernées avec `\rotatebox{angle}{texte}` (pour l'utiliser, ajouter `\usepackage{graphicx}` dans l'en-tête).

	Analyse et programmation	Expression et communication	
Samuel Dupont	4.3	2.5	<pre>\begin{tabular}{c r r } &amp;\rotatebox{90}{Analyse et programmation} &amp;\rotatebox{90}{Expression et communication}  \\ \hline Samuel Dupont &amp; 4.3 &amp; 2.5\\ Manuel Dupond &amp; 3.4 &amp; 5.2\\ \end{tabular}</pre>
Manuel Dupond	3.4	5.2	

## Inclusion d'images externes sur fichier

Ajouter `\usepackage{graphicx}` dans l'en-tête du document et utiliser la commande

`\includegraphics[options]{fichier}`

pour afficher l'image `fichier` avec certaines options.

Quelques exemples pour les options (au besoin, séparer par des virgules) :

`height=5cm` pour ramener la hauteur de l'image à 5cm  
`width=8cm` pour ramener la largeur de l'image à 8cm  
`angle=45` pour tourner l'image de 45°

Si la hauteur ou la largeur est spécifiée, mais pas les deux, l'autre grandeur est adaptée de manière à ce que les proportions originales de l'image soient conservées.

## Exemple

heig-vd  
Haute Ecole d'Ingénierie et de Gestion  
du Canton de Vaud

```
\includegraphics[width=4cm,angle=15]
{logo-HEIG-VD.eps}
```

Lorsque les images sont nombreuses, il est judicieux de les regrouper dans un ou plusieurs répertoire(s) dont le chemin peut être préalablement spécifié à l'aide de la commande suivante :

```
\graphicspath{{rep1/}{rep2/}...{repn/}}
```

## Attention

Si vous compilez (voir page 11) avec

- `latex` tous les fichiers d'images doivent être au format `.eps` ou `.ps`,
- `pdflatex` tous les fichiers d'images doivent être au format `.pdf`, `.jpg` ou `.png`.

Au besoin, certaines images pourront être converties dans un format adéquat à l'aide d'un logiciel tel que

ImageMagick  
GIMP  
Photoshop  
...

## Les objets flottants

Généralement, les **tableaux** et **figures** sont des **objets flottants** qui sont **numérotés** et dotés d'une **description** à l'aide des environnements

```
\begin{table}[pos]          \begin{figure}[pos]
  Un_tableau                Une_figure
  \caption{Description}      \caption{Description}
\end{table}                 \end{figure}
```

L'ordre des lettres dans l'argument optionnel [pos] indique des **préférences**

- [h] laisser l'objet **là où il se trouve** dans le source
- [t] placer l'objet **en haut** d'une page (page courante ou la plus proche)
- [b] placer l'objet **en bas** d'une page (page courante ou la plus proche)

La commande optionnelle `\caption{ }` permet de spécifier la légende du flottant (au dessus ou au dessous suivant où elle est placée) et peut contenir un `\label{ }` pour faire une référence à son numéro ailleurs dans le texte.

## Exemples flottants

voiture	bus
camion	avion

Table 18: Transports

Texte entre la table 18 et  
la figure de la page 58.



Figure 1: Logo

```
\setcounter{table}{17}
\begin{table}[ht]
\begin{center}
  \begin{tabular}{c|c}
    voiture & bus\\ \hline
    camion & avion
  \end{tabular}
\caption{\label{TableTransp}Transports}
\end{center}
\end{table}
Texte entre la table \ref{TableTransp}
et la f{igure} de la page \pageref{FigLogo}.
\begin{figure}[hb]
\begin{center}
  \includegraphics[width=3cm]
    {logo-HEIG-VD.eps}
  \caption{\label{FigLogo}Logo}
\end{center}
\end{figure}
```

## Liste des figures et des tables

De manière analogue à la table des matières, il est possible d'afficher une liste des figures

### Table des figures

1	Logo . . . . .	58	\listoffigures
2	Un cercle . . . . .	87	
3	Un cercle . . . . .	88	

... et une liste des tables apparaissant dans le document

### Liste des tableaux

18	Transports . . . . .	58	\listoftables
----	----------------------	----	---------------

## Partie 5

### Tableaux et formules mathématiques



## Tableaux de largeur donnée

L'environnement `tabularx` (inclure le package dans l'en-tête) fonctionne comme `tabular` sauf que l'on spécifie une **largeur totale** pour le tableau et que les colonnes affublées du descripteur spécial `X` **se partagent** l'espace horizontal disponible.

Exemple	
x	3
18	Cet environnement joue sur la largeur des colonnes

```
\begin{tabularx}{5cm}{|c|X|}
\hline
\multicolumn{2}{|c|}{Exemple}\\
\hline
x & 3\\
\hline
18 & Cet environnement joue
sur la largeur des colonnes\\
\hline
\end{tabularx}
```

## Tableaux sur plusieurs pages

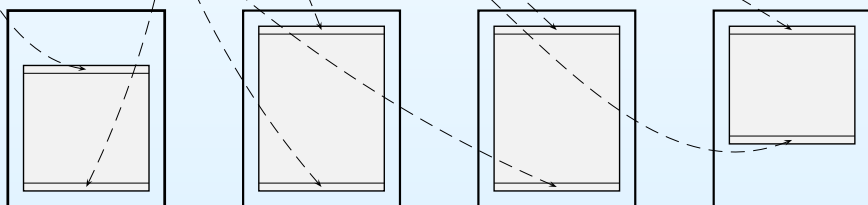
Le package `longtable` (à inclure dans l'en-tête) sert à représenter des tableaux qui s'étendent sur plusieurs pages. Bien que le tableau de l'exemple ci-dessous soit plus court que cela, il suffit à illustrer la syntaxe des principales fonctionnalités de cet environnement :

Symboles chimiques	
Symbole	Nom
C	Carbone
Au	Or
Zn	Zinc

```
\begin{longtable}[c]{|c|l|}
\hline\multicolumn{2}{|c|}{Symboles chimiques}\\
\hline
Symbole & Nom\\
\hline
C & Carbone\\
Au & Or\\
Zn & Zinc\\
\hline
\end{longtable}
```

## Tableaux sur plusieurs pages II

- `\endfirsthead` indique la fin de la série de lignes à afficher au tout début du tableau.
- `\endhead` indique la fin de la série de lignes à afficher sur la partie du tableau située en haut de chaque nouvelle page.
- `\endlastfoot` indique la fin de la série de lignes à afficher à la fin du tableau.
- `\endfoot` indique la fin de la série de lignes à afficher sur la partie du tableau terminant une page.



## Formules mathématiques

On dénombre trois principaux modes d'écriture d'expressions mathématiques :

– Les équations **centrées numérotées** avec l'environnement `equation`

$$(1) \quad \int_0^3 x^2 dx = 9$$

```
\begin{equation}
\int_0^3 x^2 dx=9
\end{equation}
```

– Les équations **centrées non numérotées** avec l'environnement `\[...\]`

$$\int_0^3 x^2 dx = 9$$

```
\[
\int_0^3 x^2 dx=9
\]
```

– Dans le **texte** les équations sont **condensées** et à placer entre deux `$$...$$`

Sachant que  $\int_0^3 x^2 dx = 9$ , on en conclut

Sachant que `$$\int_0^3 x^2 dx=9$`, on en conclut

## Remarques

- Comme pour les sections, il est possible de placer un `\label{}` dans une equation et de faire **référence** à son numéro avec `\ref{}` ou à la page où elle se situe avec `\pageref{}`.
- Par défaut, pour un document de la classe `article`, les équations sont **numérotées** séquentiellement à travers tout le document. Par contre, avec la classe `book` ou `report`, les équations sont numérotées par chapitre.
- Les commandes `\textstyle` et `\displaystyle` permettent de **forcer ou d'annuler la mise en forme condensée** des équations :

(2)  $\int_0^3 x^2 dx = 9$

Sachant que  $\int_0^3 x^2 dx = 9$ , on en conclut que la limite...

```
\begin{equation}\textstyle
\int_0^3 x^2 dx=9
\end{equation}
Sachant que $\displaystyle
\int_0^3 x^2 dx=9$, on
en conclut que la limite...
```

## Indices et exposants

- Les **indices** s'obtiennent à l'aide de la commande `_{\indice}`

$$x_1 + x_2 + x_{y_1+y_2} = 1$$

```
\[
x_{1} + x_2 + x_{y_1+y_2}=1
\]
```

- Si l'indice/exposant ne comprend qu'un caractère, on peut supprimer les {}.
- Les **exposants** s'obtiennent à l'aide de la commande `^{\exposant}`

$$x^1 + x^2 + x^{y_1+y_2} = 1$$

```
\[
x^{1} + x^2 + x^{y_1+y_2}=1
\]
```

- On peut bien sûr combiner et imbriquer les deux de manière quelconque

$$x_b^{z_{tot}+y_{x_2+x_3}^a}$$

```
\[
x_b^{z_{tot}+y^{a}_{x_2+x_3}}
\]
```

## Fractions et racines

- Pour obtenir des **fractions**, on utilise `\frac{num}{denom}`

$$\frac{2x+1}{3x-2} + \frac{\frac{2}{x-1} + \frac{5}{7x+6}}{\frac{2}{x-1} + \frac{5}{7x+6}}$$

```
\[
\frac{2x+1}{3x-2} + \frac{\frac{2}{x-1} + \frac{5}{7x+6}}{\frac{2}{x-1} + \frac{5}{7x+6}}
\]
```

- Plus simplement, si l'on souhaite obtenir  $2/3$ , il suffit d'écrire `$2/3$`.

- **Racines** : on utilise `\sqrt[exp]{val}`, où `[exp]` est optionnel

$$\sqrt[3]{7x} + \sqrt{1 + \sqrt{y + \sqrt[7]{1+x}}}$$

```
\[
\sqrt[3]{7x} + \sqrt{1 + \sqrt{y + \sqrt[7]{1+x}}}
\]
```

## Symboles I

L<sup>A</sup>T<sub>E</sub>X comprend tellement de **symboles et opérateurs** mathématiques qu'il n'est pas envisageable de les énumérer ici. Nous renvoyons donc le lecteur à la documentation pour un inventaire. En voici quelques-uns des plus courants :

$\infty$	<code>\infty</code>	$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>
$\emptyset$	<code>\varnothing</code>	$\emptyset$	<code>\emptyset</code>	$\neg$	<code>\neg</code>
$'$	<code>\prime</code>	$\parallel$	<code>\parallel</code>	$\angle$	<code>\angle</code>
$\flat$	<code>\flat</code>	$\natural$	<code>\natural</code>	$\sharp$	<code>\sharp</code>
$\pm$	<code>\pm</code>	$\setminus$	<code>\setminus</code>	$\wr$	<code>\wr</code>
$\cap$	<code>\cap</code>	$\cup$	<code>\cup</code>	$\times$	<code>\times</code>
$\vee$	<code>\vee</code>	$\wedge$	<code>\wedge</code>	$\cdot$	<code>\cdot</code>
$\triangleleft$	<code>\triangleleft</code>	$\circ$	<code>\circ</code>	$\bullet$	<code>\bullet</code>
$\triangleright$	<code>\triangleright</code>	$\div$	<code>\div</code>	$\amalg$	<code>\amalg</code>

## Symboles II

$\in$	<code>\in</code>	$\ni$	<code>\ni</code>	$\notin$	<code>\notin</code>
$\partial$	<code>\partial</code>	$\Re$	<code>\Re</code>	$\Im$	<code>\Im</code>
$\triangle$	<code>\triangle</code>	$\imath$	<code>\imath</code>	$\jmath$	<code>\jmath</code>
$\diamond$	<code>\diamond</code>	$\oplus$	<code>\oplus</code>	$\odot$	<code>\odot</code>
$\bot$	<code>\bot</code>	$\top$	<code>\top</code>	$\nabla$	<code>\nabla</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$\doteq$	<code>\doteq</code>
$\leq$	<code>\leq</code>	$\geq$	<code>\geq</code>	$\equiv$	<code>\equiv</code>
$\prec$	<code>\prec</code>	$\succ$	<code>\succ</code>	$\sim$	<code>\sim</code>
$\preceq$	<code>\preceq</code>	$\succeq$	<code>\succeq</code>	$\simeq$	<code>\simeq</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\neq$	<code>\neq</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>
$\asymp$	<code>\asymp</code>	$\propto$	<code>\propto</code>	$\models$	<code>\models</code>
$\checkmark$	<code>\checkmark</code>	$\Box$	<code>\Box</code>	$\mid$	<code>\mid</code>

## Symboles III

La **négation** de certains symboles s'obtient en les faisant précéder de `\not`

$\not<$	<code>\not&lt;</code>	$\not>$	<code>\not&gt;</code>	$\not\equiv$	<code>\not\equiv</code>
$\not\leq$	<code>\not\leq</code>	$\not\geq$	<code>\not\geq</code>	$\not\cong$	<code>\not\cong</code>
$\not\subset$	<code>\not\subset</code>	$\not\supset$	<code>\not\supset</code>	...	

Poursuivons avec quelques **flèches**...

$\leftarrow$	<code>\leftarrow</code>	$\rightarrow$	<code>\rightarrow</code>
$\Lleftarrow$	<code>\Lleftarrow</code>	$\Rrightarrow$	<code>\Rrightarrow</code>
$\longleftarrow$	<code>\longleftarrow</code>	$\longrightarrow$	<code>\longrightarrow</code>
$\Llongleftarrow$	<code>\Llongleftarrow</code>	$\Rlongrightarrow$	<code>\Rlongrightarrow</code>
$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\mapsto$	<code>\mapsto</code>
$\Llongleftrightarrow$	<code>\Llongleftrightarrow</code>	$\longmapsto$	<code>\longmapsto</code>
$\nearrow$	<code>\nearrow</code>	$\searrow$	<code>\searrow</code>

## Lettres grecques

$\alpha$	<code>\alpha</code>	$\omega$	<code>\omega</code>	$\pi$	<code>\pi</code>
$\beta$	<code>\beta</code>	$\lambda$	<code>\lambda</code>	$\mu$	<code>\mu</code>
$\gamma$	<code>\gamma</code>	$\upsilon$	<code>\upsilon</code>	$\nu$	<code>\nu</code>
$\delta$	<code>\delta</code>	$\kappa$	<code>\kappa</code>	$\xi$	<code>\xi</code>
$\rho$	<code>\rho</code>	$\varrho$	<code>\varrho</code>	$\eta$	<code>\eta</code>
$\phi$	<code>\phi</code>	$\varphi$	<code>\varphi</code>	$\tau$	<code>\tau</code>
$\sigma$	<code>\sigma</code>	$\varsigma$	<code>\varsigma</code>	$\chi$	<code>\chi</code>
$\theta$	<code>\theta</code>	$\vartheta$	<code>\vartheta</code>	$\iota$	<code>\iota</code>
$\epsilon$	<code>\epsilon</code>	$\varepsilon$	<code>\varepsilon</code>	$\psi$	<code>\psi</code>
$\zeta$	<code>\zeta</code>	$\Upsilon$	<code>\Upsilon</code>	$\Psi$	<code>\Psi</code>
$\Theta$	<code>\Theta</code>	$\Lambda$	<code>\Lambda</code>	$\Xi$	<code>\Xi</code>
$\Gamma$	<code>\Gamma</code>	$\Delta$	<code>\Delta</code>	$\Pi$	<code>\Pi</code>
$\Sigma$	<code>\Sigma</code>	$\Omega$	<code>\Omega</code>	$\Phi$	<code>\Phi</code>

## Parenthésage

Pour que dans une formule la **taille des délimiteurs** (parenthèses ( ), accolades `\{ \}`, crochets `[ ]`, barres verticales `| |` et bien d'autres) **s'adapte automatiquement à la hauteur du contenu**, il faut faire précéder ces éléments des commandes `\left` et `\right` :

$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)f(x,y)$	<code display="block">\left[\begin{array}{l} (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})f(x,y) \\ f(x,y) \end{array}\right]</code>
$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)f(x,y)$	<code display="block">\left[\begin{array}{l} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)f(x,y) \\ f(x,y) \end{array}\right]</code>

## Paranthésage manuel

Hormis la taille de base et le mode automatique, il est possible de spécifier une taille explicitement en préfixant le nom du délimiteur par

`\big`, `\Big`, `\bigg` ou `\Bigg`

Par exemple

$()\{\}$	<pre>\[   () \{\}</pre>
$()\{\}$	<pre>\[   \Big(\Big) \Big\{\Big\}</pre>
$()\{\}$	<pre>\[   \Bigg(\Bigg) \Bigg\{\Bigg\}</pre>

Cette possibilité **n'est** généralement **pas utilisée** (sauf cas particulier).

## Divers I

– Les **parties entières** inférieures et supérieures

$\left\lfloor \frac{p}{q} \right\rfloor \leq \frac{p}{q} \leq \left\lceil \frac{p}{q} \right\rceil$	<pre>\[ \left\lfloor \frac{p}{q} \right\rfloor \leq \frac{p}{q} \leq \left\lceil \frac{p}{q} \right\rceil</pre>
---	---

– Les **ensembles**  $\mathbb{N}$ ,  $\mathbb{Z}$ ,  $\mathbb{Q}$ ,  $\mathbb{R}$  et  $\mathbb{C}$  s'obtiennent en utilisant `\mathbb{bb}{}`

$\forall x \in \mathbb{R}$  `\$forall x\in\mathbb{bb}{R}$`

Il est nécessaire d'inclure le package `amssymb` pour utiliser `\mathbb{bb}{}`.

– On forme les majuscules **calligraphiques**  $\mathcal{A}$ ,  $\mathcal{B}$ ,... avec `\mathcal{}`

$\exists \alpha \in \mathcal{P}$  `\$exists\alpha\in\mathcal{P}$`

## Partie 6

### Formules mathématiques

## Divers II

• Quelques **pointillés**...

$\cdots$  `\cdots` |  $\ldots$  `\ldots` |  $\vdots$  `\vdots` |  $\ddots$  `\ddots`

• Quelques **accents** mathématiques

$\vec{a}$ <code>\vec{a}</code>	$\dot{a}$ <code>\dot{a}</code>	$\check{a}$ <code>\check{a}</code>
$\bar{a}$ <code>\bar{a}</code>	$\hat{a}$ <code>\hat{a}</code>	$\tilde{a}$ <code>\tilde{a}</code>

• Utiliser `\underbrace{ }_{}{}` pour ajouter une **accolade au dessous** d'une formule (et `\overbrace{ }^{}{}` pour la placer au dessus) :

$\underbrace{ax^2 + bx + c}_{f(x)} = 0$	<code>\[ \underbrace{ax^2+bx+c}_{f(x)}=0 \]</code>
---	--

• Utiliser `\mbox` pour **sortir du mode mathématique** et inclure du texte :

$k \in \mathbb{Z}$ avec $f(k) = 0$	<code>\\$k\in\mathbb{bb}{Z}\mbox{ avec }f(k)=0\$</code>
------------------------------------	---

## Divers III

– Les coefficients **binomiaux** s'obtiennent avec `{haut\choose bas}`

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$$

– Pour placer un **objet sur un autre**, on a `\stackrel{haut}{bas}`

$$A \xrightarrow{\phi} B$$

– Deux formes d'utilisation du **modulo** :

On a  $x \equiv y \pmod{b}$ ,  
mais aussi  $a = b \bmod c$

## Sommes, produits, intégrales,...

Les **grands opérateurs** parmi lesquels on compte

$\sum$	$\prod$	$\coprod$
$\int$	$\bigcap$	$\bigwedge$
$\oint$	$\bigcup$	$\bigvee$

admettent des **bornes** inférieures et supérieures avec `_{\inf}` et `^{\sup}` :

$$\sum_{i=0}^{\infty} 3^{-i} = \frac{3}{2}$$

$$\int_0^3 x^2 dx = 9$$

## Fonctions usuelles

Attention à ne pas oublier le `\` devant le nom des fonction mathématiques

$$\cos x \not\equiv \cos x$$

De nombreuses **fonctions usuelles** sont reconnues par  $\text{\LaTeX}$ , notamment

<code>\arccos</code>	<code>\cos</code>	<code>\cosh</code>	<code>\inf</code>	<code>\min</code>
<code>\arcsin</code>	<code>\sin</code>	<code>\sinh</code>	<code>\sup</code>	<code>\max</code>
<code>\arctan</code>	<code>\tan</code>	<code>\tanh</code>	<code>\liminf</code>	<code>\ln</code>
<code>\arg</code>	<code>\cot</code>	<code>\coth</code>	<code>\limsup</code>	<code>\log</code>
<code>\lim</code>	<code>\exp</code>	<code>\dim</code>	<code>\deg</code>	<code>\det</code>

Certaines peuvent avoir des **informations** attachées en indice/exposant :

$$\lim_{x \rightarrow \infty} e^{-x} = 0$$

## Expressions sur plusieurs lignes

Pour un développement sur **plusieurs lignes** (chacune affublée d'un numéro d'équation), l'environnement `eqnarray` permet de travailler en mode mathématique comme dans un tableau de trois colonnes formatées avec `{rcl}`. On utilise `&` pour gérer l'alignement et `\\` pour changer de ligne :

$$\begin{eqnarray} K_n &= & \sqrt{n} \sup_{t \in I} |F_n(t) - F(t)| \\ &= & \sqrt{n} \sup_{t \in I} \left| \frac{A([0, t], x)}{n} - t \right| \end{eqnarray}$$

(3)

La commande `\nonumber` permet d'indiquer que certaines lignes ne doivent **pas être numérotées** et l'environnement alternatif `eqnarray*` fonctionne de la même manière, mais sans aucune numérotation.

## Les tableaux en mathématiques

L'environnement `array` est au mode mathématique ce que `tabular` est au mode textuel. La syntaxe est identique et il est ainsi possible d'écrire des **matrices** ou des **systèmes d'équations**. On a par exemple

```
\[
  A = \left(
    \begin{array}{ccc}
      12 & 23 & 45 \\
      27 & 64 & 13
    \end{array}
  \right), \quad
  b = \left(
    \begin{array}{c}
      3 \\
      8
    \end{array}
  \right)
\]
```

$$A = \begin{pmatrix} 12 & 23 & 45 \\ 27 & 64 & 13 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 8 \end{pmatrix}$$

## Parenthésages fantômes

Notons que si l'on ouvre une parenthèse avec `\left(`, il est nécessaire de la fermer, soit avec une autre parenthèse `\right)`, mais pourquoi pas avec un crochet `\right]`, une accolade `\right\}`, etc. Lorsqu'aucun délimiteur ne doit apparaître sur la droite, le **symbole fantôme** `\right.` est utilisé.

```
\[
  \left\{ \begin{array}{*{4}{rc}r}
    x_1 & + & x_2 & + & x_3 & + & x_4 & = & 10 \\
    2x_1 & + & x_2 & + & 2x_3 & & & = & 7 \\
    x_1 & & & + & x_3 & & & = & 2
  \end{array} \right.
\]
```

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 10 \\ 2x_1 + x_2 + 2x_3 = 7 \\ x_1 + x_3 = 2 \end{cases}$$

## Alternatives

Une autre utilisation courante de `array` est l'écriture des **alternatives** :

```
\[
  F_X(x) = P(X \leq x) =
  \left\{ \begin{array}{cl}
    0 & \text{\mbox{si } } x < 0, \\
    1 - e^{-\lambda x} & \text{\mbox{si } } x \geq 0.
  \end{array} \right.
\]
```

$$F_X(x) = P(X \leq x) = \begin{cases} 0 & \text{si } x < 0, \\ 1 - e^{-\lambda x} & \text{si } x \geq 0. \end{cases}$$

Pour terminer, mentionnons l'existence du package `amsmath` qui contient la spécification d'environnements mathématiques bien plus sophistiqués.

## Partie 7

### Graphiques avec *PSTricks*

## Les graphiques

Il est possible de créer ses graphiques à l'intérieur même de  $\text{\LaTeX}$  plutôt que d'importer des images produites à l'aide d'un logiciel extérieur. Il existe plusieurs packages pour la **création de graphiques**, mais la meilleure solution est sans doute PSTricks. Pour l'utiliser, ajouter dans l'en-tête `\usepackage{pstricks,pst-plot,pstricks-add}`.

Attention, il n'est pas possible de compiler (voir page 11) du code PSTricks avec `pdflatex`. Il faut utiliser le compilateur `latex`.

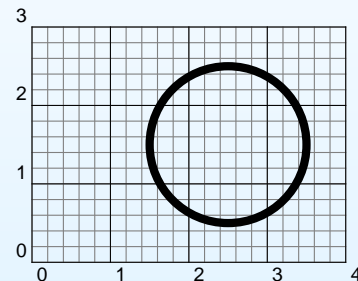
Nous nous contenterons d'un **tour d'horizon** des possibilités de PSTricks, mais nous nous attarderons un peu plus longtemps sur la génération de graphes (représentation graphique de données ou de fonctions).

Notons que des commandes PSTricks peuvent être placées directement **dans du texte**. Par exemple, la commande `\pscircle{1cm}` a été placée dans le code source juste après le point précédent...

## L'environnement pspicture

En général, on place du PSTricks dans un environnement `pspicture`. Cet environnement contient la définition d'une zone rectangulaire de travail que l'on spécifie à l'aide des coordonnées de deux sommets opposés.

La commande `\psset` permet de fixer certains paramètres graphiques (ici, l'épaisseur du trait et la longueur d'une unité en  $x$  et en  $y$ ).



```
\psset{linewidth=3pt,
       xunit=1cm,yunit=1cm}
\begin{pspicture}(0,0)(4,3)
  \psgrid
  \pscircle(2.5,1.5){1}
\end{pspicture}
```

La commande `\psgrid` (qui admet également de nombreux paramètres...) est parfois utilisée de manière temporaire, durant la création d'une figure.

## Références

En général, on inclut un environnement `pspicture` dans un environnement `figure` afin de pouvoir le **référencer**.

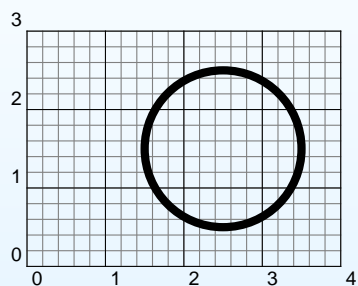


Figure 2: Un cercle

Un cercle sur la Figure 2.

```
\begin{figure}[htb]
\begin{center}
  \psset{linewidth=3pt,
         xunit=1cm,yunit=1cm}
  \begin{pspicture}(0,0)(4,3)
    \psgrid
    \pscircle(2.5,1.5){1}
  \end{pspicture}
  \caption{\label{FCercle}Un cercle}
\end{center}
\end{figure}
Un cercle sur la Figure \ref{FCercle}.
```

## Ajouter un fond

Certains aiment placer un **fond** à l'arrière des figures PSTricks. Pour cela, il suffit de placer l'environnement `pspicture` dans un `\psframebox*` { }

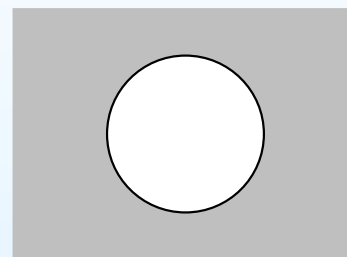


Figure 3: Un cercle

```
\begin{figure}[htb]
\begin{center}
  \psframebox*[fillcolor=lightgray]
  {
    \begin{pspicture}(0,0)(4,3)
      \pscircle[fillstyle=solid,
                fillcolor=white](2,1.5){1}
    \end{pspicture}
  }
  \caption{Un cercle}
\end{center}
\end{figure}
```

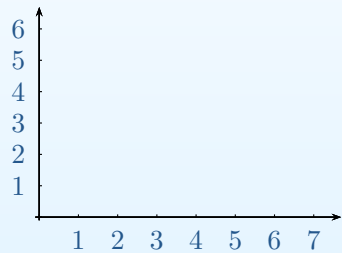


## Les axes

Avant de tracer le graphe d'une fonction, on place généralement un système d'**axes** dans la zone de travail à l'aide de la commande

```
\psaxes[options]{flèches}(Ox,Oy)(SOx,SOy)(NEx,NEy)
```

où  $(Ox,Oy)$   $(SOx,SOy)$   $(NEx,NEy)$  sont les coordonnées de l'origine, du coin sud-ouest et du coin nord-est (le 1<sup>er</sup> ou les deux 1<sup>ers</sup> points étant optionnels).



```
\psset{linewidth=0.5pt,
       xunit=0.5cm,yunit=0.4cm}
\begin{pspicture}(-1,-0.5)(7.7,6.7)
  \psaxes[labels=all,ticks=1pt]
    {->}(0,0)(-0.1,-0.1)(7.7,6.7)
\end{pspicture}
```

La liste des options étant généreuse (idem pour les flèches), nous renvoyons le lecteur à la documentation sur <http://tug.org/PSTricks/>

## Importation de données (paires de valeurs)

D'abord, les données sont à stocker dans une liste de paires de nombres

- soit en les énumérant dans le code source avec la commande  

```
\savedata{\variable}[{\{x1,y1\},{x2,y2\},...,{xn,yn\}}]
```
- soit en lisant le contenu d'un fichier externe à l'aide de la commande  

```
\readdata{\variable}{nom_du_fichier}
```

Dans ce cas, au niveau du format du fichier `nom_du_fichier`, il est possible d'utiliser des accolades {}, des parenthèses (), des sauts de ligne, des virgules et/ou des espaces. Le tout peut aussi être placé entre crochets [], ce qui a pour effet d'accélérer le processus d'importation.

Cette souplesse au niveau du format permet de récupérer la sortie de bon nombre de programmes externes dont peuvent provenir vos données.

Ainsi, le contenu du fichier pourrait par exemple être

```
(0,1)(1,0.8)(1.5,2)
```

## Variantes format d'importation

ou de manière équivalente

```
[{\{0,1\}{1,0.8}{1.5,2\}}]
```

ou

```
0,1 , 1,0.8 , 1.5,2
```

ou

```
[ 0 1    1 0.8    1.5 2 ]
```

ou

```
0 1
1 0.8
1.5 2
```

ou même

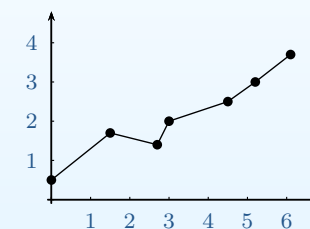
```
[{(0 1) ,
{1 , 0.8}} (1.5 2)]
```

## Graphique de mesures (en 2D)

Dans les deux cas, pour **représenter des données** sur un graphique, on utilise

```
\dataplot[options]{\variable}
```

À nouveau, la liste des options de tracé est longue (voir documentation) et nous nous contenterons de détailler deux exemples : le 1<sup>er</sup> avec `\savedata` :

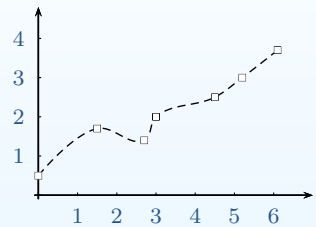


```
\savedata{\data}[{\{0,0.5\},{1.5,1.7},{2.7,1.4},
                    {3,2},{4.5,2.5},{5.2,3},{6.1,3.7\}}]
\psset{linewidth=0.5pt,xunit=0.5cm,yunit=0.5cm}
\begin{pspicture}(-0.8,-0.2)(7,4.8)
  \psaxes[ticks=1pt]{->}(0,0)(-0.1,-0.1)(7,4.8)
  \dataplot[showpoints=true,dotsize=2pt 3]{\data}
\end{pspicture}
```

Par défaut, `\dataplot` relie les points successifs par des segments (option par défaut `plotstyle=line`). En ajoutant `showpoints=true`, les points de `\data` sont mis en évidence, ici à l'aide de cercles pleins (option par défaut `dotstyle=*`) de diamètre `dotsize 2pt+3*linewidth`.

## Variante avec importation depuis un fichier

Les mêmes données, mais importées depuis `donnees.dat` avec `\readdata` :

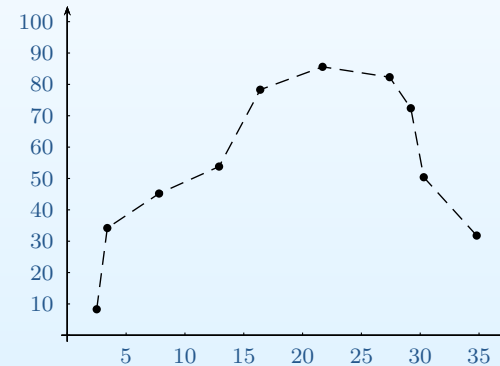


```
\readdata{\data}{donnees.dat}
\psset{linewidth=0.5pt,xunit=0.5cm,yunit=0.5cm}
\begin{pspicture}(-0.8,-0.2)(7,4.8)
\psaxes[ticksiz=1pt]{->}(0,0)(-0.1,-0.1)(7,4.8)
\dataplot[plotstyle=curve,
          linestyle=dashed,dash=3pt 2pt,
          showpoints=true,dotstyle=square]{\data}
\end{pspicture}
```

L'option `plotstyle=curve` fait passer une courbe par les différents points de `\data`. L'option `linestyle=dashed` indique que l'on veut un tracé en traitillé et `dash=3pt 2pt` permet de préciser que l'on souhaite un motif de traitillé avec 3pt de trait suivi de 2pt de vide. Avec `showpoints=true`, les points de `\data` sont à nouveau mis en évidence, mais cette fois-ci à l'aide de carrés vides (`dotstyle=square`) avec option par défaut pour leur taille.

## Partant de deux colonnes de mesures .xls

- Supprimer tout, hormis les données numériques à exporter
- Enregistrer en format texte (`mesures.txt` ou `mesures.csv`) en choisissant l'espace, la tabulation ou la virgule comme séparateur
- Adapter le code suivant (estimer `xunit`, `yunit`, la taille du `psaxes` et du `pspicture`, ainsi que les unités sur les axes `Dx` et `Dy`) :



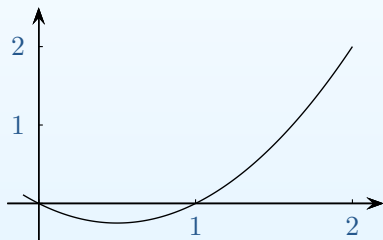
```
\readdata{\data}{mesures.csv}
\psset{linewidth=.5pt,
       xunit=0.15cm,yunit=0.04cm}
\begin{pspicture}(-3,0)(38,105)
\psaxes[ticksiz=1pt,Dx=5,Dy=10]
      {->}(0,0)(-0.5,-2)(38,105)
\dataplot[linestyle=dashed,
          showpoints=true]{\data}
\end{pspicture}
```

## Grapher d'une fonction $y = f(x)$

Il est aussi possible de tracer le **graphe**  $y = f(x)$  d'une fonction donnée par son expression mathématique (pour  $x \in [x_{\min}, x_{\max}]$ ) avec

```
\psplot[algebraic, autres options]{xmin}{xmax}{f(x)}
```

Attention,  $f$  doit forcément être exprimée **en fonction de la variable  $x$** .



```
\psset{linewidth=0.5pt,xunit=2cm,yunit=1cm}
\begin{pspicture}(-0.1,-0.5)(2.1,2.5)
\psaxes[arrowscale=1.8,ticksiz=1.5pt]
      {->}(0,0)(-0.2,-0.5)(2.2,2.5)
\psplot[algebraic,plotpoints=200]{-0.1}{2}
      {x^2-x}
\end{pspicture}
```

Le paramètre optionnel `plotpoints` permet de fixer le nombre de points en lesquels la fonction est évaluée. Plus ce nombre est élevé, plus la courbe est lisse.

## Exprimer la fonction

Les principales **opérations mathématiques** à disposition sont :

+	^	abs()	EXP()	round()
-	cos()	sqrt()	ACOS()	floor()
*	sin()	ln()	ASIN()	ceiling()
/	tan()	log()	ATAN()	truncate()

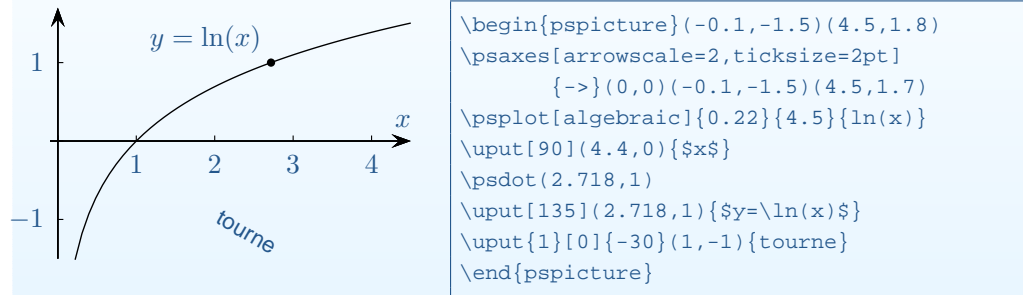
Notons que les fonctions trigonométriques fonctionnent en **radians**.

Quelques exemples :

$\sqrt{3x}$	<code>sqrt(3*x)</code>
$2^x$	<code>2^x</code>
$x^2 + 5x - 7$	<code>x^2+5*x-7</code>
$e^{5x}$	<code>EXP(5*x)</code>

## Ajouter du texte ou une formule sur une figure

La commande `\uput[angle](x,y){objet}` permet de placer un objet (par exemple du texte ou une formule) dans la **direction** donnée par **angle** (en degrés) autour du **point de référence** de coordonnées  $(x,y)$ .



En cas de besoin, la syntaxe plus générale

`\uput{distance}[angle]{rotation}(x,y){objet}` permet de placer l'objet à une certaine distance du point de référence  $(x,y)$  (dans la direction **angle**) et de le tourner d'un angle **rotation**.

## Courbes paramétrées

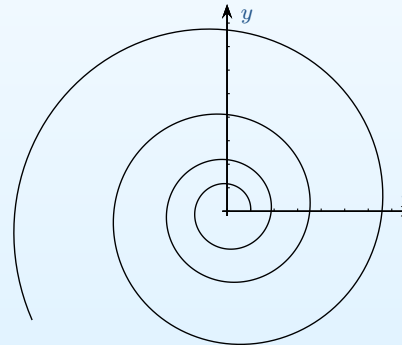
Pour une **courbe paramétrée** dans le plan  $\{(x(t), y(t)), t \in [t_{\min}, t_{\max}]\}$ , on spécifie **deux fonctions** à séparer à l'aide du symbole `|`

`\parametricplot[algebraic]{tmin}{tmax}{x(t) | y(t)}`

et qui doivent être exprimées à l'aide de la variable **t** (plutôt que **x**).

Par exemple, pour la spirale

$$\begin{cases} x(t) = e^{t/10} \cos t \\ y(t) = e^{t/10} \sin t \end{cases} \text{ pour } t \in [0, 22.5]$$



```
\begin{pspicture}(-8.8,-5)(8,7.2)
\psaxes[ticks=1pt,arrowscale=1.5,
labels=none]{->}(0,0)(-0.2,-0.2)(8,8.8)
\parametricplot[algebraic,plotpoints=500]
{0}{22.5}{EXP(t/10)*cos(t) |
EXP(t/10)*sin(t)}
\uput[90](7.7,0){$x$}
\uput[0](0,8.3){$y$}
\end{pspicture}
```

## Autres graphiques

De nombreuses autres formes de graphiques peuvent être créés à l'aide de **PSTricks**, par exemple pour réaliser des graphiques en trois dimensions ou pour représenter des résultats statistiques : histogrammes, diagrammes en camembert, etc.

Il suffit pour cela d'utiliser des packages tels que `pst-3dplot`, `pstricks-add`, `pst-bar`, etc.

## Partie 8

### *PSTricks par l'exemple*

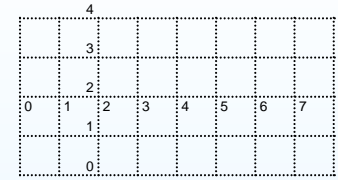
## Remarque

PSTricks est un outil de dessin offrant de larges possibilités. Pour chaque commande, les **options sont nombreuses** et il n'est pas question de les inventorier ici. Notre objectif est d'effectuer un **tour d'horizon** et de présenter une **sélection** de fonctionnalités courantes illustrées par des **exemples**.

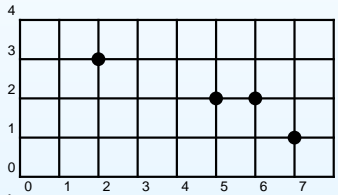
Pour plus de détails sur les commandes de base ou sur les nombreuses extensions disponibles, il suffit de consulter la **documentation** disponible sur le site de PSTricks :

<http://tug.org/PSTricks/>

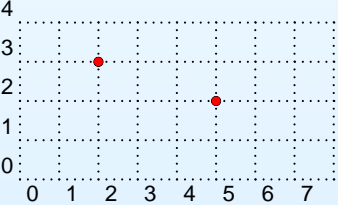
## Grilles (\psgrid) et points (\psdots)



```
\begin{pspicture}(0,0)(8,4)
\psgrid[subgriddiv=1,griddots=10]
(2,2)(0,0)(8,4)
\end{pspicture}
```

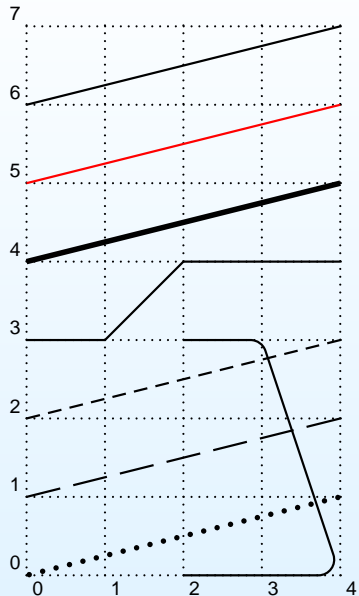


```
\begin{pspicture}(0,0)(8,4)
\psgrid[griddots=0,gridwidth=0.8pt](0,0)(8,4)
\psdots[dotsize=5pt](2,3)(5,2)(7,1)(6,2)
\end{pspicture}
```



```
\begin{pspicture}(0,0)(8,4)
\psgrid[griddots=5,gridlabels=8pt](0,0)(8,4)
\psdots[dotstyle=o,fillcolor=red](2,3)(5,2)
\end{pspicture}
```

## Lignes avec \psline



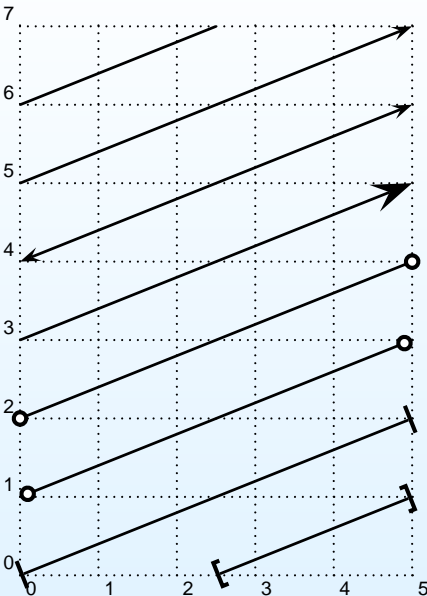
```
\begin{pspicture}(-0.3,0)(4,7)
\psgrid[subgriddiv=1,griddots=10]

\psline(0,6)(4,7)
\psline[linecolor=red](0,5)(4,6)
\psline[linewidth=2pt](0,4)(4,5)

\psline(0,3)(1,3)(2,4)(4,4)
\psline[linearc=0.2](2,3)(3,3)(4,0)(2,0)

\psline[linestyle=dashed](0,2)(4,3)
\psline[linestyle=dashed,dash=13pt 5pt]
(0,1)(4,2)
\psline[linestyle=dotted,linewidth=2pt]
(0,0)(4,1)
\end{pspicture}
```

## Flèches et autres extrémités



```
\begin{pspicture}(0,0)(5,7)
\psgrid[subgriddiv=1,griddots=10]

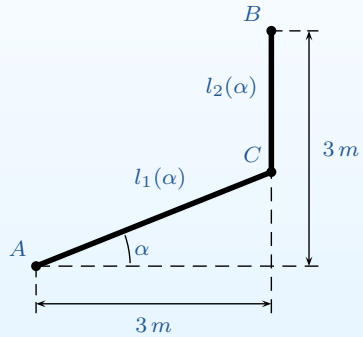
\psline(0,6)(2.5,7)

\psline{->}(0,5)(5,7)
\psline{<->}(0,4)(5,6)
\psline[arrowscale=3]{->}(0,3)(5,5)

\psline{o-o}(0,2)(5,4)
\psline{oo-oo}(0,1)(5,3)

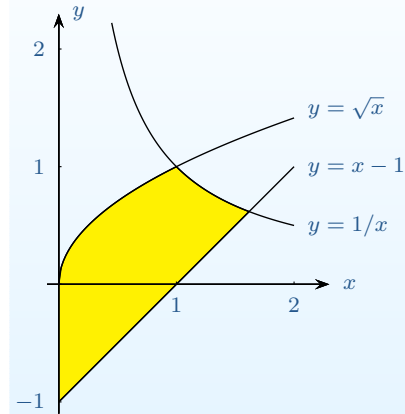
\psline{|-|}(0,0)(5,2)
\psline{[-]}(2.5,0)(5,1)
\end{pspicture}
```

## Exemple : résistivité



```
\begin{pspicture}(-1,-1)(6,5)
\psline[linewidth=2pt](0,0)(5,2)(5,5)
\psdot(0,0)\uput[135](0,0){$A$}
\psdot(5,5)\uput[135](5,5){$B$}
\psdot(5,2)\uput[135](5,2){$C$}
\psset{linewidth=0.5pt}
\psline[linestyle=dashed](0,0)(6,0)
\psline[linestyle=dashed](0,0)(0,-1)
\psline[linestyle=dashed](5,-1)(5,5)
\psline[linestyle=dashed](5,5)(6,5)
\psarc[arcsep=1pt](0,0){1.2}{1}{20}
\uput[20](1.8,0.2){$\alpha$}
\uput[180](5,3.8){$l_2(\alpha)$}
\uput[225](3.4,2.3){$l_1(\alpha)$}
\psline{<->}(0,-0.8)(5,-0.8)
\uput[270](2.5,-0.8){$3\,m$}
\psline{<->}(5.8,0)(5.8,5)
\uput[0](5.8,2.5){$3\,m$}
\end{pspicture}
```

## Colorier \pscustom

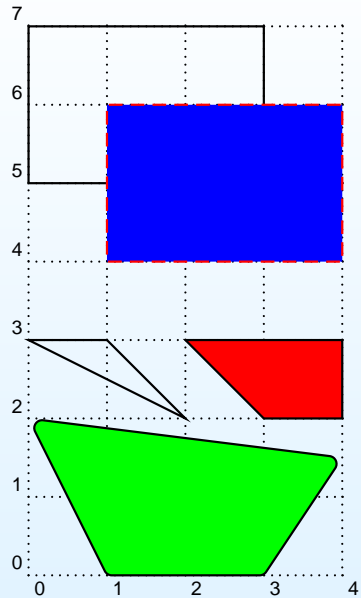


```
\begin{pspicture}(-0.1,-1.1)(2.5,2.5)
\pscustom[fillstyle=solid,fillcolor=yellow]
{
\psline(0,-1)(1.618,0.618)
\psplot[algebraic]{1.618}{1}{1/x}
\psplot[algebraic]{1}{0}{sqrt(x)}
}

\psline(0,-1)(2,1)
\psplot[algebraic]{0.45}{2}{1/x}
\psplot[algebraic]{0}{2}{sqrt(x)}

\psaxes[ticksizel=1pt,arrowscale=1.5]
{->}(0,0)(-0.1,-1.1)(2.3,2.3)
\uput[0](2,0.5){$y=1/x$}
\uput[0](2,1){$y=x-1$}
\uput[0](2,1.5){$y=\sqrt{x}$}
\uput[0](2.3,0){$x$}
\uput[0](0,2.3){$y$}
\end{pspicture}
```

## Rectangles \psframe, Polygones \pspolygon

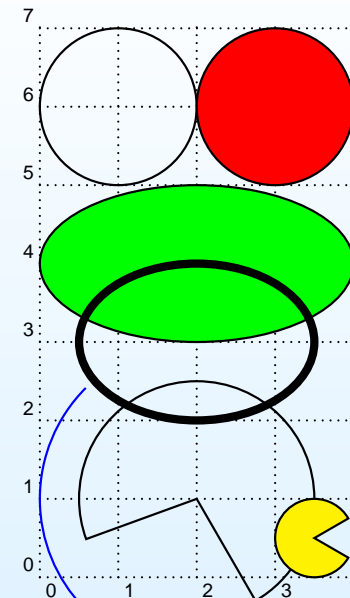


```
\begin{pspicture}(-0.3,0)(4,7)
\psgrid[subgriddiv=1,griddots=10]

\psframe(0,5)(3,7)
\psframe[fillstyle=solid,fillcolor=blue,
linestyle=dashed,linecolor=red]
(1,4)(4,6)

\pspolygon(2,2)(1,3)(0,3)
\pspolygon[fillstyle=solid,fillcolor=red]
(3,2)(2,3)(4,3)(4,2)
\pspolygon[fillstyle=solid,fillcolor=green,
lineararc=0.1]
(0,2)(1,0)(3,0)(4,1.5)
\end{pspicture}
```

## Cercles, ellipses, arcs, quartiers



```
\begin{pspicture}(-0.1,0)(4,7)
\psgrid[subgriddiv=1,griddots=10]

\pscircle(1,6){1}
\pscircle[fillstyle=solid,fillcolor=red]
(3,6){1}

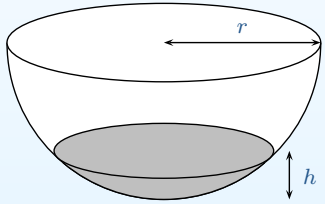
\psellipse[fillstyle=solid,fillcolor=green]
(2,4)(2,1)
\psellipse[linewidth=3pt](2,3)(1.5,1)

\psarc[linecolor=blue]{->}(2,1){2}{135}{270}

\pswedge(2,1){1.5}{300}{200}
\pswedge[fillstyle=solid,fillcolor=yellow]
(3.5,0.5){0.5}{30}{330}

\end{pspicture}
```

## Exemple : cuve semi-sphérique



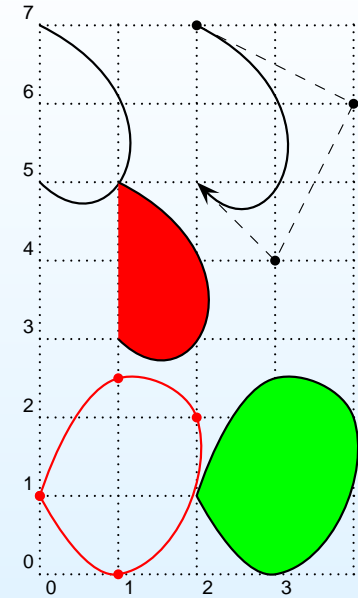
```
\begin{pspicture}(0,1)(4,4)
\psset{fillstyle=solid,
fillcolor=lightgray}

% Cuve :
\psarc[fillcolor=white](2,3){2}{180}{0}
\psellipse[fillcolor=white](2,3)(2,0.5)

% Liquide au fond
\psarc(2,3){2}{230}{310}
\psellipse(2,1.62)(1.4,0.35)

% Rayon et hauteur
\psline{<->}(2,3)(4,3)
\psline{<->}(3.6,1)(3.6,1.62)
\uput[90](3,2.95){$r$}
\uput[0](3.6,1.3){$h$}
\end{pspicture}
```

## Courbes

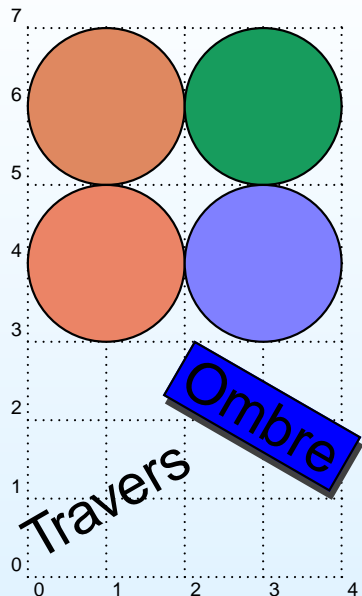


```
\begin{pspicture}(-0.1,0)(4,7)
\psgrid[subgriddiv=1,griddots=10]

\psbezier(0,7)(2,6)(1,4)(0,5) % 4 points !
\psbezier[showpoints=true,arrowscale=2]
{->}(2,7)(4,6)(3,4)(2,5)
\psbezier[fillstyle=solid,fillcolor=red]
(1,5)(3,4)(2,2)(1,3)

\pscurve[showpoints=true,linecolor=red]
(0,1)(1,2.5)(2,2)(1,0)(0,1)
\pscurve[fillstyle=solid,fillcolor=green]
(2,1)(3,2.5)(4,2)(3,0)(2,1)
\end{pspicture}
```

## Nouvelles couleurs et placement d'objets (\rput)



```
\begin{pspicture}(-0.1,0)(4,7)\black
\psgrid[subgriddiv=1,griddots=10](4,7)

\newcmkcolor{rougepale}{0 0.5 0.6 0.1}
\newcmkcolor{vertpale}{0.8 0 0.8 0.15}
\newrgbcolor{saumon}{0.92 0.52 0.4}
\newrgbcolor{mauve}{0.5 0.5 1}

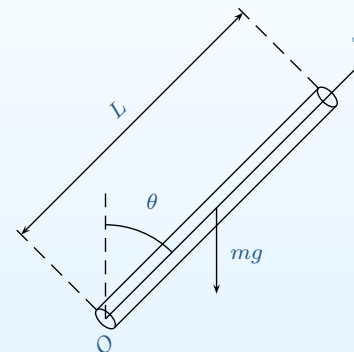
\psset{fillstyle=solid}
\pscircle[fillcolor=rougepale](1,6){1}
\pscircle[fillcolor=vertpale](3,6){1}
\pscircle[fillcolor=saumon](1,4){1}
\pscircle[fillcolor=mauve](3,4){1}

\rput{30}(1,1){\Huge Travers}

\rput{330}(3,2){\psshadowbox[fillcolor=blue]
{\Huge Ombre}}

\end{pspicture}
```

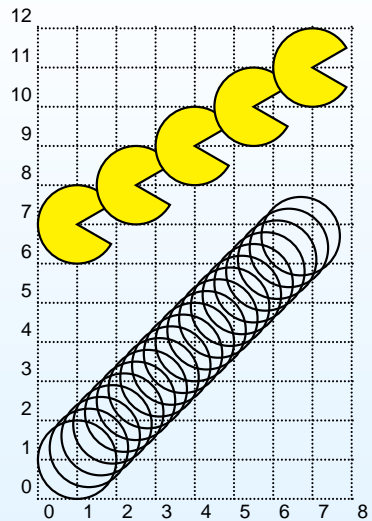
## Exemple : point de rupture



```
\begin{pspicture}(-2,0)(4.5,5)
\rput{45}(0,0){
\psline{>}(0,0)(6,0)
\psline(0,0.2)(5,0.2)
\uput[90](6,0){$r$}
\uput[225](0,0){$O$}
\psline(0,-0.2)(5,-0.2)
\psellipse(0,0)(0.1,0.2)
\psellipse(5,0)(0.1,0.2)
\psline[linestyle=dashed](0,0.2)(0,2)
\psline[linestyle=dashed](5,0.2)(5,2)
\psline{<->}(0,1.9)(5,1.9)
\uput[90](2.5,1.9){$L$}}
\psline[linestyle=dashed](0,0)(0,2)
\psline{>}(1.77,1.77)(1.77,0.4)
\uput[0](1.77,1){$mg$}
\psarc(0,0){1.2}{45}{90}
\uput[45](0.5,1.6){$\theta$}
\end{pspicture}
```



## Copies et boucles : ajouter package multido



Dans du texte, on a par exemple  
 $k = 10, k = 12, k = 14, \dots$

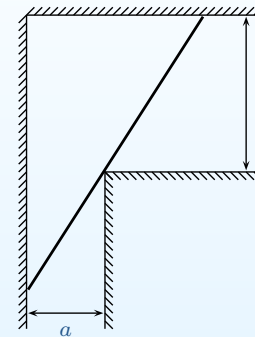
```
\begin{pspicture}(-0.1,0)(8,12)
\psgrid[subgriddiv=1,griddots=10](8,12)

% (point initial)(translation){nombre}{...}
\multips(1,7)(1.5,1){5}{
\pswedge[fillstyle=solid,fillcolor=yellow]
{0.5}{30}{330}}

\multido{\n=1+0.3}{20}{
\pscircle(\n,\n){0.5}}
% Nom de variable en n... ou N... si
% l'on ne travaille pas sur les entiers
\end{pspicture}

\medskip\footnotesize
Dans du texte, on a par exemple
\multido{\i=10+2}{3}{\k=\i$, \,}...
% Nom de variable en i ou I sur les entiers
```

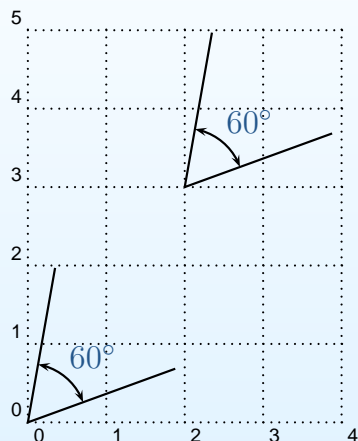
## Exemple : couloir



```
\begin{pspicture}(-0.3,-1)(3.2,-5)
\psline(0,-1)(3,-1)
\multips(0,-1)(0.1,0){30}{
\psline(0,0)(0.1,0.1)}
\psline(0,-1)(0,-5)
\multips(0,-1)(0,-0.1){40}{
\psline(0,0)(-0.1,-0.1)}
\psline(1,-3)(1,-5)
\multips(1,-3)(0,-0.1){20}{
\psline(0,0)(0.1,-0.1)}
\psline(1,-3)(3,-3)
\multips(1,-3)(0.1,0){20}{
\psline(0,0)(0.1,-0.1)}
\psline[linewidth=1pt]
(0.02,-4.5)(2.25,-1.02)
\psline{<->}(0.02,-4.8)(0.98,-4.8)
\uput[270](0.5,-4.8){$a$}
\psline{<->}(2.8,-1.02)(2.8,-2.98)
\uput[0](2.8,-2){$b$}
\end{pspicture}
```

## Coordonnées polaires

Il est parfois plus pratique d'exprimer certaines choses en coordonnées polaires  
 (rayon ; angle) plutôt qu'en coordonnées cartésiennes (x, y).



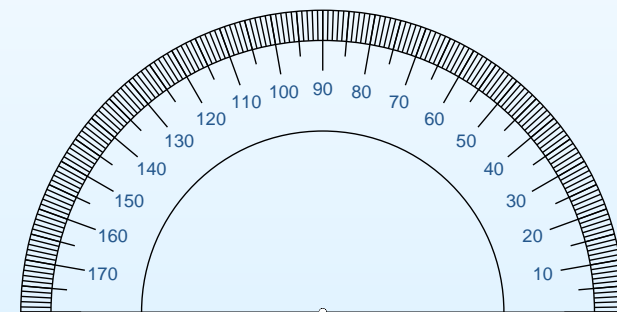
```
\begin{pspicture}(-0.1,0)(4,5)
\psgrid[subgriddiv=1,griddots=10](4,5)

\psline(2;20)(0,0)(2;80)
\psarc{<->}(0,0){0.75}{20}{80}
\uput[50](0.7;50){$60^\circ$}

\rput{0}(2,3)
{
\psline(2;20)(0,0)(2;80)
\psarc{<->}(0,0){0.75}{20}{80}
\uput[50](0.7;50){$60^\circ$}
}
\end{pspicture}
```

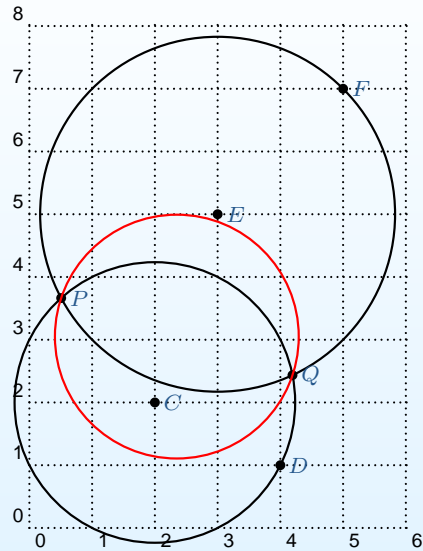
## Exemple : rapporteur

```
\begin{pspicture}(-7,-0.5)(7,5)
\psarc(0,0){5}{0}{180} \psarc(0,0){4.5}{0}{180} \psarc(0,0){3}{0}{180}
\psline(-5,0)(5,0) \psdot[dotstyle=o](0,0)
\multido{\i=0+1}{181}{\psline(4.5;\i)(5;\i)} % graduation 0,1,...,180
\multido{\i=0+10}{19}{\psline(4;\i)(5;\i)} % graduation dizaines
\multido{\i=5+10}{18}{\psline(4.25;\i)(5;\i)} % graduation 5,15,...,175
\multido{\i=10+10}{17}{\rput(3.7;\i){\tiny\i}} % texte : 10,20,...,170
\end{pspicture}
```





## Géométrie plane : ajouter package pst-eucl



```
\begin{pspicture}(-0.5,-0.5)(6,8.5)
\psgrid[subgriddiv=1,griddots=10](6,8)

% Placement des 4 points C, D, E et F
\pstGeonode(2,2){C}(4,1){D}
(3,5){E}(5,7){F}

% Cercle de centre C passant par D
% Cercle de centre E passant par F
\pstCircleOA{C}{D}
\pstCircleOA{E}{F}

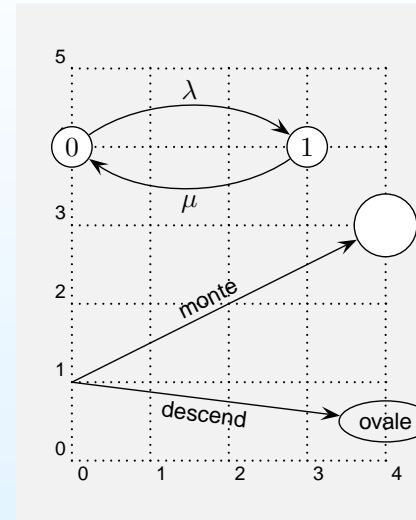
% P et Q = intersection des 2 cercles
\pstInterCC{C}{D}{E}{F}{P}{Q}

% Cercle rouge de diametre PQ
\pstCircleAB[linecolor=red]{P}{Q}
\end{pspicture}
```

## Partie 9

### Réaliser une présentation

## Théorie des graphes : ajouter package pst-node



```
\psframebox*[framesep=18pt]{
\begin{pspicture}(0,-0.2)(4,5.2)\black
\psgrid[subgriddiv=1,griddots=10](4,5)

\psset{fillcolor=white}
\cnodeput[fillstyle=solid](0,4){a}{\$0\$}
\cnodeput[fillstyle=solid](3,4){b}{\$1\$}
\psset{arrows=->,shortput=nab}
\ncarc[arcangle=35]{a}{b}^{\lambda\$}
\ncarc[arcangle=35]{b}{a}^{\mu\$}

\node(0,1){f}
\cnode[fillstyle=solid](4,3){0.4cm}{d}
\rput(4,0.5){\ovalnode{e}{ovale}}
\psset{nrot=:U}
\ncline{f}{d}^{\monte}
\ncline{f}{e}_{\descend}
\end{pspicture}}
```

## Réaliser une présentation

De nombreux packages ont été créés pour produire des présentations en L<sup>A</sup>T<sub>E</sub>X, notamment

- foiltex : obsolète
- seminar : obsolète
- prosper : ancienne gloire, mais désormais obsolète... ☹️
- powerdot : le challenger
- beamer : le champion

Nous allons donc nous intéresser à

beamer

# Introduction à $\text{\LaTeX}$ : réaliser une présentation avec Beamer

Eric Thiémard

heig-vd  
Haute Ecole d'Ingénierie et de Gestion  
du Canton de Vaud

Hes-so  
Haute Ecole spécialisée  
de Suisse occidentale

21 août 2014

## Documentation

Beamer est un outil puissant dont nous n'allons qu'effleurer la surface. Nous renvoyons le lecteur à la documentation pour une présentation plus détaillée...

### Du plus général au plus court

- La documentation de référence de Beamer :  
<http://mirrors.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf>
- Une présentation passablement détaillée :  
[http://research.microsoft.com/en-us/um/people/saikat/ref/beamer\\_guide.pdf](http://research.microsoft.com/en-us/um/people/saikat/ref/beamer_guide.pdf)
- Une présentation en ligne en français :  
<http://www.siteduzero.com/informatique/tutoriels/creez-vos-diaporamas-en-latex-avec-beamer>
- Une présentation nettement plus compacte :  
<http://web.mit.edu/rsi/www/pdfs/beamer-tutorial.pdf>
- Une courte introduction sur wikibooks :  
<http://en.wikibooks.org/wiki/LaTeX/Presentations>

## Table des matières

### 1 Structure d'un document Beamer

- En-tête du document
- Corps du document : frames
- Transitions

### 2 Définition de blocks

### 3 Animations

- La pause
- Les listes et énumérations
- Les spécifications

### 4 L'environnement `columns`

## Structure globale et options

```
\documentclass[options]{beamer}
    En-tête du document

\begin{document}
    Corps du document
\end{document}
```

### Options

La liste des options envisageables est longue (voir la documentation), par exemple la taille de base de la police utilisée : 10pt, 11pt ou 12pt.

### Pour imprimer

Dans une présentation orale, certaines pages peuvent être décomposées en plusieurs couches (certains éléments sont dévoilés progressivement). La version destinée à être imprimée s'obtient en ajoutant l'option `handout`.

## Thèmes et couleurs

En général, les premières lignes de l'en-tête indiquent le thème et le jeu de couleurs à utiliser pour donner un certain aspect à la présentation.

### Sélection d'un thème et d'un jeu de couleurs

La sélection s'effectue avec les commandes `\usetheme{NomDuTheme}` et `\usecolortheme{JeudeCouleurs}`

### Exemple : l'aspect de cette présentation a été obtenu avec

```
\usetheme{JuanLesPins}  
\usecolortheme{rose}
```

Un aperçu des principales combinaisons est accessible *via*

<http://www.hartwork.org/beamer-theme-matrix/>

Voir également

<http://mcclnews.free.fr/latex/beamergalerie/completsgalerie.html>

5/26

## Table des matières et position courante

Comme dans tout document  $\text{\LaTeX}$ , il est toujours possible d'invoquer la commande `\tableofcontents` pour faire apparaître la table des matières à n'importe quel endroit d'un document.

### Position courante

Beamer offre également la possibilité de faire apparaître automatiquement, à chaque début de section (ou de sous-section), une table des matières avec mise en évidence de la position courante.

### Le cas échéant, le code à ajouter dans l'en-tête est

```
\AtBeginSection[]{  
  \begin{frame}{Table des matières}  
    \tableofcontents[currentsection]  
  \end{frame}}
```

7/26

## Packages

Naturellement, d'autres packages (déjà rencontrés) sont généralement ajoutés dans l'en-tête

### Exemple

```
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage{lmodern}  
  
\usepackage[frenchb]{babel}  
\uselanguage{French}  
\languagepath{French}  
  
% etc.
```

6/26

## Définition des éléments de la page de titre

Typiquement, l'en-tête peut aussi contenir la définition des éléments suivants :

- le titre `\title[titre abrégé optionnel]{titre principal}`,
- l'auteur `\author{nom de l'auteur}`,
- l'institution `\institute{nom de l'institution}`,
- la date `\date{la date}`.

### Exemple pour cette présentation

```
\title[Réaliser une présentation avec Beamer]  
      {Introduction à \LaTeX{} :\\  
       réaliser une présentation avec Beamer}  
\author{Eric Thiémarc}  
\institute{\includegraphics[height=.7cm]{logo-HEIG-VD.eps}  
           \quad  
           \includegraphics[height=.7cm]{logo-HES-SO.eps}}  
\date{\today}
```

8/26

## L'environnement frame

Page de titre (optionnelle) : affiche les éléments définis dans l'en-tête

```
\begin{frame}  
  \titlepage  
\end{frame}
```

Page normale (une page par environnement frame)

```
\begin{frame}{Titre de la page (peut être vide)}  
  Contenu de la page  
\end{frame}
```

Sections et sous-sections

Les commandes de structuration du document

```
\section{...} et \subsection{...}
```

sont à placer entre les pages (en dehors des environnements frame).

9/26

## Autres transitions

Il en existe d'autres et certaines admettent des options (voir doc) :

```
\transblindshorizontal  
\transblindsvertical  
  \transboxin  
  \transboxout  
\transdissolve  
\transglitter  
\transsplitverticalin  
\transsplitverticalout  
\transsplithorizontalin  
\transsplithorizontalout  
  \transwipe
```

Pour passer à la page suivante, nous allons utiliser

```
\transdissolve
```

11/26

## Les transitions

Il s'agit d'effets utilisables (avec modération) pour passer d'une page à une autre.

Pour arriver à cette page, l'effet a été obtenu en plaçant la commande

```
\transboxout
```

juste après la commande `\begin{frame}{Les transitions}`.

De même, pour passer à la page suivante, nous allons utiliser

```
\transblindshorizontal
```

10/26

## Table des matières

- 1 Structure d'un document Beamer
  - En-tête du document
  - Corps du document : frames
  - Transitions
- 2 Définition de blocks
- 3 Animations
  - La pause
  - Les listes et énumérations
  - Les spécifications
- 4 L'environnement `columns`

12/26

# Les blocks

Beamer offre la possibilité de mettre des éléments en évidence en les plaçant dans des blocks qu'il est possible de nommer ou pas. Il en existe de trois types, associés à des couleurs différentes :

## Un block normal

En bleu, c'est beaucoup mieux...

```
\begin{block}{Un block normal}
  En bleu, c'est beaucoup mieux...
\end{block}
```

## Un block d'exemple

Je suis un exemple...

```
\begin{exampleblock}{Un block d'exemple}
  Je suis un exemple...
\end{exampleblock}
```

## Un block d'alerte

Ne pas oublier de fermer vos blocks en sortant !

```
\begin{alertblock}{Un block d'alerte}
  Ne pas oublier de fermer vos blocks
  en sortant !
\end{alertblock}
```

13/26

# Affichage progressif

Il est possible de dévoiler progressivement le contenu d'une page en utilisant la commande `\pause`.

- Il est aussi possible d'utiliser cette
- commande dans une énumération ou une liste

ou pour afficher un block

## Le code correspondant est

```
Il est possible \pause
de dévoiler progressivement le contenu d'une page \pause
en utilisant la commande \verb+\pause+. \pause
\begin{itemize}
\item Il est aussi possible d'utiliser cette \pause
\item commande dans une énumération ou une liste
\end{itemize} \pause
\begin{block}{ou pour afficher un block}\end{block}
```

15/26

# Table des matières

- 1 Structure d'un document Beamer
  - En-tête du document
  - Corps du document : frames
  - Transitions
- 2 Définition de blocks
- 3 Animations
  - La pause
  - Les listes et énumérations
  - Les spécifications
- 4 L'environnement `columns`

14/26

# Raccourci et variante

- L'option `[<+>]`, utilisable pour les listes et les énumérations est un raccourci
- qui permet d'éviter de placer des commandes `\pause` avant chaque `\item`.

```
\begin{itemize}[<+>]
\item L'option \texttt{[<+>]},
      utilisable pour les listes
      et les énumérations est un
      raccourci
\item qui permet d'éviter de
      placer des commandes
      \verb+\pause+ avant chaque
      \verb+\item+.
\end{itemize}
```

Il existe aussi une variante avec mise en évidence successive des lignes :

- A
- L
- E
- R
- T

```
\begin{itemize}[<alert@+>]
\item A
\item L
\item E
\item R
\item T
\end{itemize}
```

16/26

## Définition d'une spécification

Les animations présentées jusqu'ici sont des cas particuliers qui se laissent généraliser à l'aide de la notion de spécification.

### Spécification

Une spécification `<...>` permet de préciser quelles couches d'une page sont concernées par un affichage ou une action.

### Voici quelques exemples :

- `<2-4>` désigne les couches 2 à 4
- `<3->` désigne les couches à partir de 3
- `<4>` désigne la couche 4 uniquement
- `<1-2,4>` désigne les couches 1, 2 et 4

## Utilisation dans une liste ou une énumération

Pour chaque item d'une liste ou d'une énumération, il est possible d'utiliser une spécification pour indiquer dans quelles couches il doit apparaître.

- désigne les couches 2 à 4
- désigne les couches à partir de 3
- désigne la couche 4 uniquement
- désigne les couches 1, 2 et 4

```
\begin{itemize}
\item<2-4> désigne les
    couches 2 à 4
\item<3-> désigne les
    couches à partir de 3
\item<4> désigne la
    couche 4 uniquement
\item<1-2,4> désigne les
    couches 1, 2 et 4
\end{itemize}
```

## Autres utilisations des spécifications

### Deux animations envisageables (il y en a bien d'autres...)

- `\onslide<...>{QuelqueChose}` permet de préciser sur quelles couches le QuelqueChose (texte, tableau, block, etc.) doit apparaître.
- `\color<...>{couleur}{Texte}` permet de préciser sur quelles couches le Texte doit passer dans la couleur spécifiée.

### Remarque

En omettant le dernier paramètre des commandes ci-dessus, c'est-à-dire avec

`\onslide<...>` et `\color<...>{couleur}`,

l'action s'applique sur tout ce qui suit la commande en question.

## Exemple

Attention, mesdames et messieurs,  
le spectacle va bientôt commencer.  
D'un coup de baguette magique, je  
vais colorier ce

LAPIN

en rouge...

Abracadabra !

Yeah !

Attention, mesdames et  
messieurs, le spectacle va  
bientôt commencer.

```
\onslide<2-> D'un coup de
baguette magique, je vais
colorier ce
\begin{center}
\color<3->{red}{LAPIN}
\end{center}
en rouge...
```

```
\onslide<3->
\begin{alertblock}
{Abracadabra !}{Yeah !}
\end{alertblock}
```

## Raccourci pour les blocks

Avec des blocks, plutôt que d'écrire

Second

Premier

```
\onslide<3->{\begin{block}{  
    Second  
    \end{block}}  
\onslide<2->{\begin{block}{  
    Premier  
    \end{block}}
```

il est aussi possible d'utiliser la forme compacte

Second

Premier

```
\begin{block}<6->{  
    Second  
}\end{block}  
\begin{block}<5->{  
    Premier  
}\end{block}
```

21/26

## Faire apparaître un tableau colonne après colonne

A	1	2	3
B	4	5	6
C	7	8	9

Inclure le package colortbl et utiliser

```
\begin{tabular}{|c<{\onslide<2->}|c<{\onslide<3->}  
               c<{\onslide<4->}c<{\onslide}|}\hline  
A & 1 & 2 & 3\\  
B & 4 & 5 & 6\\  
C & 7 & 8 & 9\\ \hline  
\end{tabular}
```

23/26

## Faire apparaître un tableau ligne après ligne

A	1	2
B	3	4
C	5	6

Utiliser

```
\begin{tabular}{|c|cc|}\hline  
A & & 1 & & 2\\  
\onslide<2->{B}& \onslide<2->{3}& \onslide<2->{4}\\  
\onslide<3->{C}& \onslide<3->{5}& \onslide<3->{6}\\ \hline  
\end{tabular}
```

22/26

## Table des matières

- 1 Structure d'un document Beamer
  - En-tête du document
  - Corps du document : frames
  - Transitions
- 2 Définition de blocks
- 3 Animations
  - La pause
  - Les listes et énumérations
  - Les spécifications
- 4 L'environnement columns

24/26



## Théorie par l'exemple...

L'environnement `columns` permet de subdiviser l'espace en plusieurs colonnes, par exemple pour juxtaposer du texte, des images ou des blocks.

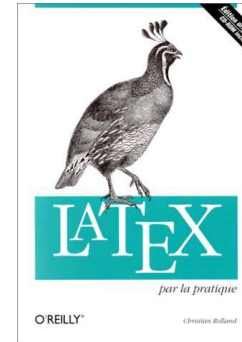
## Le code

```
\begin{columns}
  \column{0.3\textwidth}
    \includegraphics[height=4.1cm]{pic/LatexRoland.eps}
  \column{0.4\textwidth}
    \begin{alertblock}<2->{Petite page de pub}
      Ces deux livres sont en français
    \end{alertblock}
    \begin{block}<3->{Remarque}
      Le second est une traduction de la version anglaise
    \end{block}
  \column{0.3\textwidth}
    \includegraphics[height=4.1cm]{pic/Goossens.eps}
\end{columns}
```

25/26

## Partie 10

## Extensions et sélection d'éléments avancés

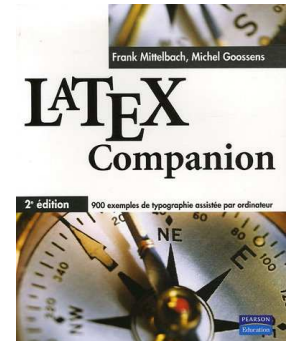
Exemple `columns`

## Petite page de pub

Ces deux livres sont en français

## Remarque

Le second est une traduction de la version anglaise



26/26

## Inclusion de fichiers

Un document est généralement scindé en plusieurs fichiers, si bien qu'il est usuel de compiler un fichier principal ne contenant qu'une en-tête, un environnement `document` et des **inclusions** de fichiers. Il existe deux formes d'inclusion :

- `\input{Nom_Fichier}` : intègre le fichier `Nom_Fichier`.
- `\include{Nom_Fichier}` : intègre le fichier `Nom_Fichier` en commençant une nouvelle page.

Par exemple, en dehors des images, le code source de ce cours comprend douze fichiers : un fichier principal, un fichier pour les trois pages d'introduction et un fichier pour chacune des dix parties.

## L'environnement minipage

L'environnement `minipage` permet de glisser de petites pages de largeur quelconque (ici de 3.5cm) dans une page usuelle. Cet environnement permet par exemple de juxtaposer une image et un paragraphe de texte.



```
L'environnement \texttt{minipage}
permet de glisser de petites pages
de largeur
\begin{minipage}[b]{3.5cm}
quelconque (ici de 3.5cm) dans
une page usuelle. Cet environnement
permet par exemple de juxtaposer une
\end{minipage}\hfill
\includegraphics[height=1.9cm]
{pic/don.eps}
image et un paragraphe de texte.
```

Le paramètre entre crochets (ici `[b]`) permet de préciser quelle partie du paragraphe doit être alignée avec la ligne courante : le bas `[b]`, le haut `[t]` ou le centre `[c]`.

Cet exemple fait également intervenir la commande `\hfill` qui remplit l'espace disponible de la ligne courante par un espace de largeur maximale.

## Nouveaux compteurs

Il est parfois utile de créer de nouveaux compteurs à l'aide de la commande `\newcounter` et d'accéder à leur valeur avec `\value` :

Normalement, chaque `enumerate`

1. commence avec la valeur 1.
2. Si l'on termine un `enumerate`,

mais que l'on souhaite reprendre la suite dans un nouvel environnement (sans partir de 1), on peut stocker la valeur terminale du premier `enumi` dans un nouveau compteur temporaire,

3. puis initialiser le compteur `enumi` du
4. 2e `enumerate` en recopiant cette valeur

```
Normalement, chaque enumerate
\begin{enumerate}
\item commence avec la valeur 1.
\item Si l'on termine un enumerate,
\newcounter{temporaire}
\setcounter{temporaire}{\value{enumi}}
\end{enumerate}
mais que l'on souhaite reprendre la suite dans
un nouvel environnement (sans partir de 1), on
peut stocker la valeur terminale du premier
enumi dans un nouveau compteur temporaire,
\begin{enumerate}
\setcounter{enumi}{\value{temporaire}}
\item puis initialiser le compteur enumi du
\item 2e enumerate en recopiant cette valeur
\end{enumerate}
```

## Nouvelles longueurs et mesures

Introduisons une nouvelle longueur et attribuons-lui une valeur en mesurant la largeur d'un texte que nous pouvons aussi visualiser : le nom « Haddock » mesure 25.75458pt en largeur.

En la diminuant de 1pt et en reculant de cette longueur, nous obtenons l'effet suivant : « Le capitaine **Haddock** a-t-il bu ? ».

Il est aussi possible de mesurer la hauteur d'un texte au dessus (4.674pt) et au dessous (0.07538pt) du niveau de la ligne de base.

```
Introduisons une nouvelle longueur
\newlength{\taille}
et attribuons-lui une valeur en mesurant la
\settowidth{\taille}{Haddock}
largeur d'un texte que nous pouvons aussi
visualiser : le nom <<~Haddock~>> mesure
\the\taille{} en largeur.\ll[0.5cm]
\addtolength{\taille}{-1pt}
En la diminuant de 1pt et en reculant de
cette longueur, nous obtenons l'effet suivant
: <<~Le capitaine
Haddock\hspace{-\taille}Haddock
a-t-il bu ?~>>.\ll[0.5cm]
\settoheight{\taille}{Haddock}
Il est aussi possible de mesurer la
hauteur d'un texte au dessus (\the\taille)
\settodepth{\taille}{Haddock}
et au dessous (\the\taille) du
niveau de la ligne de base.
```

## Redéfinition de commandes

Il est possible de redéfinir des commandes existantes. Examinons quelques cas où la commande à redéfinir n'admet pas de paramètre ; il suffit alors d'utiliser `\renewcommand{NomDeCommande}{NouveauCode}`.

Par exemple, dans une liste le symbole affiché (en `prosp`) est une boule

- default symbol

```
\begin{itemize}
\item default symbol
\end{itemize}
```

En redéfinissant la commande `\labelitemi` (ou `\labelitemii` pour une sous-liste, etc.) qui affiche ce symbole, il est facile d'afficher autre chose :

▷ Vive les triangles !

```
\renewcommand{\labelitemi}
{$\triangleright$}
\begin{itemize}
\item Vive les triangles !
\end{itemize}
```

## Redéfinition de commandes II

Reconsidérons le cas de la table des matières (`\tableofcontents`, voir page 34). Imaginons que nous souhaitons remplacer le nom par défaut (pour un document en français) « Table des matières », par « Sommaire » qui peut s'avérer plus approprié dans certains contextes.

Pour cela, il suffit de redéfinir la commande `\contentsname` :

```
\renewcommand{\contentsname}{Sommaire}
```

De même pour changer le libellé de la « Table des figures » et de la « Liste des tableaux » (voir page 59), il suffit d'utiliser

```
\renewcommand{\listfigurename}{Sommaire figures}
\renewcommand{\listtablename}{Sommaire tableaux}
```

## Nouvelles commandes sans paramètre

Il est facile de définir une nouvelle commande sans paramètre en utilisant `\newcommand{\NomMacro}{Code}`. Par exemple

Les exercices du cours L<sup>A</sup>T<sub>E</sub>X sont disponibles sur  
<http://sachem.heig-vd.ch/latex>

```
\newcommand{\SiteWeb}{
\url{http://sachem.heig-vd.ch/latex}}

Les exercices du cours LATEX sont
disponibles sur \SiteWeb
```

Les principaux avantages d'une telle macro sont

- que si l'adresse du site web change, il suffit de modifier la macro pour que toutes les occurrences dans le document soient modifiées d'un seul coup ;
- c'est plus court à taper (sans parler de l'esthétique et de la réduction du risque d'erreur...)

## Redéfinition de commande avec paramètres

Lorsque la commande à redéfinir admet `NbParam` paramètres, on utilise `\renewcommand{NomDeCommande}[NbParam]{NouveauCode}` où les paramètres sont nommés #1, #2, etc. dans `NouveauCode`.

Par exemple, la commande `\textit{chaîne}` admet une chaîne de caractères en paramètre et a pour effet d'écrire cette chaîne en italique :

C'est le *statu quo* pour nous.

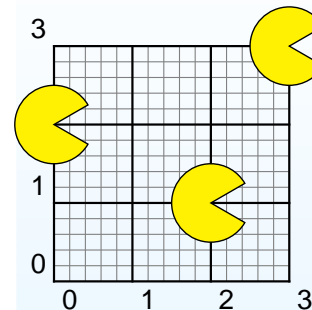
```
C'est le \textit{statu quo} pour nous.
```

Pour faire passer en rouge et en gras les chaînes en italique à partir d'un certain point du document, il suffit de redéfinir la commande `\textit` comme suit :

C'est le **statu quo** pour nous.

```
\renewcommand{\textit}[1]{\textbf{\red #1}}
C'est le \textit{statu quo} pour nous.
```

## Nouvelles commandes avec paramètres



```
\newcommand{\PacMan}[2]{
\put(#1,#2){
\pswedge[fillcolor=yellow]{0.5}{30}{330}}}
\begin{pspicture}(0,0)(3,3)
\psgrid
\PacMan{2}{1}
\PacMan{3}{3}
\PacMan{0}{2}
\end{pspicture}
```

Nom<sup>moins long</sup>

a<sup>b</sup> toto<sup>titi</sup>tutu

```
\newcommand{\expo}[1]{\textsuperscript{#1}}
Nom\expo{moins long}
```

```
\newcommand{\doubleexpo}[3]{#1\expo{#2}\expo{#3}}
\doubleexpo{a}{b}{c}
\doubleexpo{toto}{titi}{tutu}
```

Le nombre de paramètres de telles macros est limité à 9.

## Quelques packages recommandés

Certains outils classiques n'ont pas été inclus dans ce cours, par exemple

- `bibtex` pour le gestion d'une bibliographie,
- `makeindex` pour la création d'un index.

Nous avons également omis la présentation de nombreux packages fréquemment utiles, par exemple

- `fancyhdr` qui permet la paramétrisation des pieds et en-têtes de page,
- `listings` pour la publication de code source dans divers langages,
- `calc` pour effectuer des opérations arithmétiques sur des compteurs/longueurs,
- `ifthen` pour effectuer des tests ou une boucle while,
- `version` pour compiler uniquement certaines parties d'un document (par exemple inclure/exclure d'un coup les corrigés d'un support de cours).

## CTAN : Comprehensive TeX Archive Network

Le site

<http://www.ctan.org>

est **LA** bibliothèque des packages L<sup>A</sup>T<sub>E</sub>X.

On y trouve de tout...

Les fonctions de recherche et la documentation en ligne sont bien pratiques pour repérer les packages susceptibles de vous intéresser.

Bonne visite !