

Formation L^AT_EX — Niveau avancé

Vincent Fevrier

CIES

10 avril 2009

<http://www.math.u-psud.fr/~fevrier/enseignement/2008/CIES>

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
 - Manipulation de fichiers PostScript et PDF
 - Listings et programmes informatiques
 - Partitions de musique
 - Lettres et courrier
 - Affiches et posters
 - Les extensions pstricks
 - Transparents avec Seminar
 - Présentations avec Beamer
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
 - Manipulation de fichiers PostScript et PDF
 - Listings et programmes informatiques
 - Partitions de musique
 - Lettres et courrier
 - Affiches et posters
 - Les extensions pstricks
 - Transparents avec Seminar
 - Présentations avec Beamer
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Changer la taille du papier d'un fichier PostScript

Le programme `psresize` permet de changer la taille de papier utilisée pour l'impression.

Changer la taille du papier d'un fichier PostScript

Le programme `psresize` permet de changer la taille de papier utilisée pour l'impression.

Usage

```
psresize [-wWidth] [-hHeight] [-pPaperSize] Input.ps Output.ps
```

- `Width` et `Height` permettent de spécifier la taille du papier de sortie (par exemple `-w10in`);
- `PaperSize` définit une taille de papier standard (par exemple `a4` ou encore `letter`);
- les mêmes options mais en majuscule (`-W`, `-H` ou encore `-P`) permettent de définir la taille du papier d'entrée (si différente de `a4`).

Plusieurs pages par feuille avec psnup

Le programme `psnup` permet de rassembler plusieurs pages d'un document PostScript par feuille.

Plusieurs pages par feuille avec psnup

Le programme `psnup` permet de rassembler plusieurs pages d'un document PostScript par feuille.

Usage

```
psnup -p[PaperSize] -[NumberPerPage] [-bMargin] Input.ps Output.ps
```

- `PaperSize` définit la taille de papier qui sera utilisée (par exemple `a4` ou encore `letter`);
- `NumberPerPage` est le nombre de pages du document original qu'on souhaite disposer sur chaque page du document final (par exemple 2, 4 ou 8);
- `Margin` permet d'ajouter une marge supplémentaire autour des pages du document original (éventuellement négative).

Changer l'ordre des pages pour une reliure centrale avec psbook

Avant `psnup` on peut utiliser `psbook` qui va changer l'ordre des pages :



Changer l'ordre des pages pour une reliure centrale avec psbook

Avant psnup on peut utiliser psbook qui va changer l'ordre des pages :



Usage

```
psbook Input.ps Output.ps
```

Concaténer des fichiers PostScript

S'il est installé, on peut utiliser `psmerge` :

Usage

```
psmerge [-oOutput.ps] Input1.ps Input2.ps ...
```

Concaténer des fichiers PostScript

S'il est installé, on peut utiliser `psmerge` :

Usage

```
psmerge [-oOutput.ps] Input1.ps Input2.ps ...
```

Sinon avec GhostScript :

Usage

```
gs -dBATCH -dNOPAUSE -q -sDEVICE=pswrite -sOutputFile=Output.ps Input1.ps  
Input2.ps ...
```

▷ Sous Windows, remplacer `gs` par `gswin32c` (dans le répertoire `/bin` de l'installation de GhostScript).

Concaténer des fichiers PDF

Avec GhostScript :

Usage

```
gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -sOutputFile=Output.pdf Input1.pdf  
Input2.pdf ...
```

Extraire un intervalle de pages d'un document

Exemple avec `psselect`, pour extraire les pages de 5 à 10 et de 15 à 20 d'un document PostScript :

Usage

```
psselect -p5-10,15-20 Input.pdf Output.pdf
```

Extraire un intervalle de pages d'un document

Exemple avec `psselect`, pour extraire les pages de 5 à 10 et de 15 à 20 d'un document PostScript :

Usage

```
psselect -p5-10,15-20 Input.pdf Output.pdf
```

Exemple avec `GhostScript`, pour extraire les pages de 5 à 10 d'un document PDF :

Usage

```
gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -sOutputFile=Output.pdf -dFirstPage=5  
-dLastPage=10 Input.pdf
```

Vectoriser les fontes d'un fichier PostScript

... pour importer des formules dans un logiciel de dessin par exemple

Avec `pstoedit` :

Usage

```
pstoedit -dt -f ps Input.ps Output.ps
```

Vectoriser les fontes d'un fichier PostScript

... pour importer des formules dans un logiciel de dessin par exemple

Avec `pstoedit` :

Usage

```
pstoedit -dt -f ps Input.ps Output.ps
```

Exemple d'utilisation pour importer une formule en PDF dans Inkscape :

Commandes

```
latex formule.tex  
dvips formule.dvi  
pstoedit -dt -f ps formule.ps out.ps  
ps2pdf out.ps
```

La formule est alors prête à être incluse dans un logiciel de dessin, sous forme de courbes (et avec l'aspect des fontes originales de \LaTeX).

Exercice

Transformer un document \LaTeX de plusieurs pages en un fichier PostScript avec 2 pages par côté de feuille

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
 - Manipulation de fichiers PostScript et PDF
 - **Listings et programmes informatiques**
 - Partitions de musique
 - Lettres et courrier
 - Affiches et posters
 - Les extensions pstricks
 - Transparents avec Seminar
 - Présentations avec Beamer
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

L'extension listings

Trois manières d'écrire du code, d'usage similaire aux environnements verbatim :

- `\lstinline@let rec factoriel n = if n<2 then 1 else n*factoriel (n-1);;`
affichera `let rec factoriel n = if n<2 then 1 else n*factoriel (n-1);;` au milieu du texte;

L'extension listings

Trois manières d'écrire du code, d'usage similaire aux environnements verbatim :

- `\lstinline@let rec factoriel n = if n<2 then 1 else n*factoriel (n-1);;` affichera `let rec factoriel n = if n<2 then 1 else n*factoriel (n-1);;` au milieu du texte;
- `\begin{lstlisting}...\end{lstlisting}` en mode « display » donnera par exemple :

```
function Factoriel (const n: Integer): Integer;  
var  
  i: Integer;  
begin  
  Result := 1;  
  for i := 2 to n do  
    Result := Result * i;  
  end;
```

L'extension listings

Trois manières d'écrire du code, d'usage similaire aux environnements verbatim :

- `\lstinline@let rec factoriel n = if n<2 then 1 else n*factoriel (n-1);;` affichera `let rec factoriel n = if n<2 then 1 else n*factoriel (n-1);;` au milieu du texte;
- `\begin{lstlisting}... \end{lstlisting}` en mode « display » donnera par exemple :

```
function Factoriel (const n: Integer): Integer;
var
  i: Integer;
begin
  Result := 1;
  for i := 2 to n do
    Result := Result * i;
  end;
```

- `\lstinputlisting{File}` fera la même chose en affichant le contenu d'un fichier source externe.

Listings flottants

Code



Fichier Beamer3/code20.tex

```
\begin{lstlisting}[  
  uu language=C,  
  uu frame=Tb,  
  uu float,  
  uu caption=Un exemple flottant  
]  
/*_Fonction_factorielle_*/  
int_factoriel(int_n){  
  uu int_s=1;  
  uu for_(;n;--n)  
  uuuu s*=n;  
  uu return_s;  
}  
\end{lstlisting}
```

Sortie

Listing 1 – Un exemple flottant

```
/* Fonction factorielle */  
int factoriel(int n){  
  int s=1;  
  for (;n;--n)  
    s*=n;  
  return s;  
}
```

Contrôle de l'apparence

Deux manières de contrôler la façon dont le code s'affiche :

- au niveau des commandes elles-mêmes, en argument optionnel

Code



Fichier Beamer3/code22.tex

```
\begin{lstlisting}[language=html,  
  basicstyle=\itshape,keywordstyle  
  =\color{magenta}]  
<HTML><HEAD><TITLE>Ma_page_web</TITLE  
></HEAD><BODY><H1>Ma_page_web</H  
1></BODY></HTML>  
\end{lstlisting}
```

Sortie

```
<HTML><HEAD><TITLE>Ma page  
web</TITLE></HEAD><BODY>  
<H1>Ma page web</H1></  
BODY></HTML>
```

Contrôle de l'apparence

Deux manières de contrôler la façon dont le code s'affiche :

- au niveau des commandes elles-mêmes, en argument optionnel

Code



Fichier Beamer3/code26.tex

```
\begin{lstlisting}[language=html,
  basicstyle=\itshape,keywordstyle
  =\color{magenta}]
<HTML><HEAD><TITLE>Ma_page_web</TITLE>
</HEAD><BODY><H1>Ma_page_web</H1></BODY></HTML>
\end{lstlisting}
```

Sortie

```
<HTML><HEAD><TITLE>Ma page
web</TITLE></HEAD><BODY>
<H1>Ma page web</H1></
BODY></HTML>
```

- au niveau global

Code



Fichier Beamer3/code28.tex

```
\lstset{language=Matlab,numbers=left,
  numberstyle=\tiny\white}
\begin{lstlisting}
function result=factoriel(n)
if (n<=1) result=1;
else result=n*factoriel(n-1);
end;
\end{lstlisting}
```

Sortie

```
1 function result=factoriel(n)
2 if (n<=1) result=1;
3 else result=n*factoriel(n-1);
4 end;
```

Contrôle de l'apparence

Quelques options, parmi d'autres :

- `language=[dialect]language` pour spécifier le langage utilisé (par exemple `[LaTeX]TeX` ou `[ANSI]C`);

Contrôle de l'apparence

Quelques options, parmi d'autres :

- `language=[dialect]language` pour spécifier le langage utilisé (par exemple `[LaTeX]TeX` ou `[ANSI]C`);
- `basicstyle=`, `identifierstyle=`, `commentstyle=`, `stringstyle=`, `keywordstyle=`, `directivestyle=` pour le style de texte des différents éléments syntaxiques (couleur, fontes);

Contrôle de l'apparence

Quelques options, parmi d'autres :

- `language=[dialect]language` pour spécifier le langage utilisé (par exemple `[LaTeX]TeX` ou `[ANSI]C`);
- `basicstyle=`, `identifierstyle=`, `commentstyle=`, `stringstyle=`, `keywordstyle=`, `directivestyle=` pour le style de texte des différents éléments syntaxiques (couleur, fontes);
- `showspaces = <true|false>` (ou encore `showstringspaces`) pour imprimer ou non les espaces avec le symbole `␣`;

Contrôle de l'apparence

Quelques options, parmi d'autres :

- `language=[dialect]language` pour spécifier le langage utilisé (par exemple `[LaTeX]TeX` ou `[ANSI]C`);
- `basicstyle=`, `identifierstyle=`, `commentstyle=`, `stringstyle=`, `keywordstyle=`, `directivestyle=` pour le style de texte des différents éléments syntaxiques (couleur, fontes);
- `showspaces = <true|false>` (ou encore `showstringspaces`) pour imprimer ou non les espaces avec le symbole `␣`;
- `numbers = <none|left|right>` pour la numérotation des lignes;

Contrôle de l'apparence

Quelques options, parmi d'autres :

- `language=[dialect]language` pour spécifier le langage utilisé (par exemple `[LaTeX]TeX` ou `[ANSI]C`);
- `basicstyle=`, `identifierstyle=`, `commentstyle=`, `stringstyle=`, `keywordstyle=`, `directivestyle=` pour le style de texte des différents éléments syntaxiques (couleur, fontes);
- `showspaces = <true|false>` (ou encore `showstringspaces`) pour imprimer ou non les espaces avec le symbole `␣`;
- `numbers = <none|left|right>` pour la numérotation des lignes;
- `frame = <trblTRBL>` pour tracer sélectivement certains côtés du rectangle englobant le listing.

Contrôle de l'apparence

On peut définir des styles globaux :

Code



Fichier Beamer3/code30.tex

```
\lstdefinestyle{mylatex}{
  \language=[LaTeX]TeX,
  \breaklines, \showspaces,
  \basicstyle=\footnotesize\color{
    black}\ttfamily,
  \keywordstyle=\color{red},
  \commentstyle=\color{green},
  \identifierstyle=\color{blue},
  \frame=tblr,
  \numbers=left,
  \numberstyle=\color{white}
}
\begin{lstlisting}[style=mylatex,
  title=Style_mylatex]
%Un commentaire
Comment_croyez-vous_qu'a_été_défini_l
'environnement_LaTeX\qui
affiche_ceci?
\end{lstlisting}
```

Sortie

Style mylatex

1 %Un commentaire

2 Comment_croyez-vous_qu'a_été_défini_l
défini_l'environnement_\n
LaTeX\qui
affiche_ceci?

Contrôle de l'apparence

On peut définir des environnements raccourcis, sur le modèle de `\newenvironment` :

Code



Fichier Beamer3/code31.tex

```
\lstnewenvironment{mylatexenv}{  
  \lstset{style=mylatex}  
  \begin{center}---Début du code  
  ---\end{center}  
}{  
  \begin{center}---Fin du code---\  
  end{center}  
}  
\begin{mylatexenv}  
\lstnewenvironment{mylatexenv}{  
  \lstset{style=mylatex}  
  \begin{center}---Début du code  
  ---\end{center}  
}{  
  \begin{center}---Fin du code---\  
  end{center}  
}  
\end{mylatexenv}
```

Sortie

— Début du code —

```
1 \lstnewenvironment{mylatexenv}{  
2 \lstset{style=mylatex}  
3 \begin{center}---Début du  
  code---\end{center}  
4 }{  
5 \begin{center}---Fin du code  
  ---\end{center}  
6 }
```

— Fin du code —

Caractère d'échappement

L'option `escapechar=` définit un caractère permettant de revenir au mode \LaTeX , par exemple pour écrire des formules dans les commentaires.

Code



Fichier `Beamer3/code32.tex`

```
\begin{lstlisting}[language=Java,
  escapechar=@]
class Factoriel{
/* Cette fonction utilise la relation
  @\begin{equation}n!=\prod_{i=1}^n i\end{equation}@ */
static int factoriel(int n){
    int res=1;
    for(int i=1;i<=n;i++)
        res=res*i;
    return res;
}
}\end{lstlisting}
```

Sortie

```
class Factoriel{
/* Cette fonction utilise la
  relation
```

$$n! = \prod_{i=1}^n i \quad (1)$$

```
*/
static int factoriel(int n)
{
    int res=1;
    for(int i=1;i<=n;i++)
        res=res*i;
    return res;
}
}
```

Caractère d'échappement

L'option `escapechar=` définit un caractère permettant de revenir au mode \LaTeX , par exemple pour écrire des formules dans les commentaires.

Code



Fichier Beamer3/code34.tex

```
\begin{lstlisting}[language=Java,
  escapechar=@]
class Factoriel{
/* Cette fonction utilise la relation
  @\begin{equation}n!=\prod_{i=1}^n i\end{equation}@ */
  static int factoriel(int n){
    int res=1;
    for(int i=1;i<=n;i++)
      res=res*i;
    return res;
  }
}\end{lstlisting}
```

Sortie

```
class Factoriel{
/* Cette fonction utilise la
  relation
```

$$n! = \prod_{i=1}^n i \quad (1)$$

```
*/
static int factoriel(int n)
{
  int res=1;
  for(int i=1;i<=n;i++)
    res=res*i;
  return res;
}
}
```

On peut aussi utiliser `mathescape=true`, dans ce cas le symbole `$` permet de passer en mode mathématique. L'option `texcl=true` permet de passer automatiquement en mode \LaTeX dans les commentaires.

Les extensions algorithm et algorithmic

Pour écrire du pseudo-code.

Code



Fichier Beamer3/code36.tex

```
\begin{algorithm}
\caption{\operatorname{factorial}(
  n)}\label{alg:factorial}
\begin{algorithmic}[1]
\REQUIRE An integer  $n \geq 0$ .
\ENSURE The value of  $n!$ .
\medskip
\IF  $n = 0$ 
\RETURN 1
\ELSE
\RETURN  $n \cdot \operatorname{factorial}(n-1)$ 
\ENDIF
\end{algorithmic}
\end{algorithm}
```

Sortie

Algorithm 1 factorial(n)

Require: An integer $n \geq 0$.

Ensure: The value of $n!$.

- 1: if $n = 0$ then
 - 2: return 1
 - 3: else
 - 4: return $n \cdot \operatorname{factorial}(n - 1)$
 - 5: end if
-

Exercice

Écrire un « quine » en \LaTeX (c'est à dire un programme qui affiche son propre code-source).

Solution

Écrire un « quine » en \LaTeX (c'est à dire un programme qui affiche son propre code-source).

Code



Fichier Beamer3/code38.tex

```
\documentclass{article}
\usepackage{listings}
\lstset{language=[LaTeX]TeX}
\pagestyle{empty}
\begin{document}
\lstinputlisting{\jobname.tex}
\end{document}
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
 - Manipulation de fichiers PostScript et PDF
 - Listings et programmes informatiques
 - **Partitions de musique**
 - Lettres et courrier
 - Affiches et posters
 - Les extensions pstricks
 - Transparents avec Seminar
 - Présentations avec Beamer
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Écrire des partitions de musique avec l'extension musixtex

Code



Fichier Beamer3/code39.tex

```
\begin{music}
\tiny
\parindent1cm
\instrumentnumber{1}\%_a_single_instrument
\setname1{Piano}\%_whose_name_is_Piano
\setstaves1{2}\%_with_two_staves
\generalmeter{\meterfrac44}\%_4/4_meter_chosen
\startextract\%_starting_real_score
\notes\ibu0f0\qb0{cge}\tbu0\qb0g|\hl_j\en
\notes\ibu0f0\qb0{cge}\tbu0\qb0g|\ql_l\sk\ql_n\en
\bar
\notes\ibu0f0\qb0{dgf}|\qlp_i\en
\notes\tbu0\qb0g|\ibbl1j3\qb1j\tbl1\qb1k\en
\notes\ibu0f0\qb0{cge}\tbu0\qb0g|\hl_j\en
\endextract\%_terminate_excerpt
\end{music}
```

Sortie

Piano

The image shows a musical score for a piano piece titled "Sortie". The score is written in 4/4 time and consists of two staves. The right hand (treble clef) plays a melody that begins with a half note G4, followed by quarter notes A4, B4, and C5. A fermata is placed over the C5 note. The second ending, marked with a "2", starts with a quarter note D5, followed by quarter notes E5, F5, and G5. The left hand (treble clef) plays a steady accompaniment of eighth notes, alternating between G3 and A3 in the first measure, and B3 and C4 in the second measure. The word "Piano" is written to the left of the staves.

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
 - Manipulation de fichiers PostScript et PDF
 - Listings et programmes informatiques
 - Partitions de musique
 - **Lettres et courrier**
 - Affiches et posters
 - Les extensions pstricks
 - Transparents avec Seminar
 - Présentations avec Beamer
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Exemple minimaliste



Fichier Beamer3/code40.tex

```
\documentclass[français]{letter}

\usepackage[français]{babel}

\address{...\\...\\...}
\signature{...}

\begin{document}
  \begin{letter}{M...\\...\\...}
    \opening{Cher...}
    ...
    \closing{Bien cordialement}
  \end{letter}
  \begin{letter}{M...\\...\\...}
    \opening{Cher...}
    ...
    \closing{Bien cordialement}
  \end{letter}
\end{document}
```

La classe letter

Commandes supplémentaires à utiliser juste après `\closing` :

- `\cc{...}` s'il y a des destinataires en copie ;
- `\encl{...}` s'il y a des pièces jointes ;
- `\ps{PS: ...}` s'il y a un post-scriptum (ou autre).



La classe `lettre`

Elle fonctionne quasiment comme `letter`, avec quelques commandes supplémentaires, et respecte mieux les conventions françaises.

La classe lettre

Elle fonctionne quasiment comme letter, avec quelques commandes supplémentaires, et respecte mieux les conventions françaises.

On peut définir des paramètres par défaut pour toutes ses lettres.

À mettre dans un fichier `default.ins` ou juste avant `\opening` dans chaque lettre



Fichier Beamer3/code42.tex

```
%Nom et signature de l'expéditeur
\name{...}
\signature{...}
%Coordonnées de l'expéditeur
\address{...}
\location{...}
\telephone{...}% ou \notelephone
\fax{...}% ou \nofax
\email{...}% ou \noemail
%lieu d'expédition des lettres
\lieu{...}
```

La classe lettre

Commandes propres à chaque lettre

Commandes à utiliser juste avant `\opening` :

- `\conc{...}` pour définir l'objet de la lettre.

La classe lettre

Commandes propres à chaque lettre

Commandes à utiliser juste avant `\opening` :

- `\conc{...}` pour définir l'objet de la lettre.

Commandes à utiliser juste après `\closing` :

- `\cc{...}` s'il y a des destinataires en copie ;
- `\enc1{...}` s'il y a des pièces jointes ;
- `\ps{PS:}{...}` s'il y a un post-scriptum.

La classe lettre

Commandes propres à chaque lettre

Commandes à utiliser juste avant `\opening` :

- `\conc{...}` pour définir l'objet de la lettre.

Commandes à utiliser juste après `\closing` :

- `\cc{...}` s'il y a des destinataires en copie ;
- `\encl{...}` s'il y a des pièces jointes ;
- `\ps{PS:}{...}` s'il y a un post-scriptum.

Pour faire un téléfax il suffit de remplacer

```
\begin{letter}{destinataire}
```

par

```
\begin{telex}{numéro}{destinataire}
```



Exercice

Si vous ne l'avez jamais fait avec \LaTeX auparavant...

- écrire une (courte !) lettre adressée à la personne de votre choix ;

Exercice

Si vous ne l'avez jamais fait avec \LaTeX auparavant...

- écrire une (courte !) lettre adressée à la personne de votre choix ;
- la transformer en un téléfax.

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
 - Manipulation de fichiers PostScript et PDF
 - Listings et programmes informatiques
 - Partitions de musique
 - Lettres et courrier
 - **Affiches et posters**
 - Les extensions pstricks
 - Transparents avec Seminar
 - Présentations avec Beamer
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

La classe a0poster

Pour écrire sur de grandes tailles de papier avec des fontes adaptées.

La classe a0poster

Pour écrire sur de grandes tailles de papier avec des fontes adaptées.

Les options sont les suivantes :

- `a0b`, `a0`, `a1`, `a2` ou `a3` pour la taille de papier ;
- `landscape` ou `portrait` pour l'orientation.

La classe a0poster

Pour écrire sur de grandes tailles de papier avec des fontes adaptées.

Les options sont les suivantes :

- `a0b`, `a0`, `a1`, `a2` ou `a3` pour la taille de papier ;
- `landscape` ou `portrait` pour l'orientation.

La classe introduit trois tailles de fontes supplémentaires : `\veryHuge`, `\VeryHuge` et `\VERYHuge`.

L'extension textpos

Deux environnements supplémentaires pour disposer des éléments à un endroit précis :

- `\begin{textblock}{hsize}[x0,y0]` ($hpos, vpos$) permet de placer du texte dans une boîte de largeur $hsize$ dont l'origine est décalée de $(hpos, vpos)$ par rapport à la position courante.

Le décalage et la largeur sont exprimés en une unité correspondant par défaut au $1/16$ de la page.

Par défaut l'origine $(x0, y0)$ est le coin supérieur gauche de la boîte, elle peut être déplacée en donnant une fraction des dimensions de la boîte (par exemple $(0.5, 1)$ pour le milieu du bas de la boîte).

L'extension textpos

Deux environnements supplémentaires pour disposer des éléments à un endroit précis :

- `\begin{textblock}{hsize}[x0,y0]` (`hpos`, `vpos`) permet de placer du texte dans une boîte de largeur `hsize` dont l'origine est décalée de (`hpos`, `vpos`) par rapport à la position courante ;
- L'environnement `textblock*` fait la même chose, mais avec des longueurs en unités standard ;

L'extension textpos

Deux environnements supplémentaires pour disposer des éléments à un endroit précis :

- `\begin{textblock}{hsize}[x0,y0]` (`hpos`, `vpos`) permet de placer du texte dans une boîte de largeur `hsize` dont l'origine est décalée de (`hpos`, `vpos`) par rapport à la position courante ;
- L'environnement `textblock*` fait la même chose, mais avec des longueurs en unités standard ;
- `\TPGrid[marginx,marginy]{nx}{ny}` modifie les unités de façon à découper la page en $nx \times ny$ (éventuellement avec des marges).

L'extension textpos

Quelques options de l'extension :

- `showboxes` permet d'afficher le contour des boîtes ;

L'extension textpos

Quelques options de l'extension :

- `showboxes` permet d'afficher le contour des boîtes ;
- `absolute` dispose les blocs relativement au coin supérieur gauche de la page (au lieu de la position courante) ;

L'extension textpos

Quelques options de l'extension :

- `showboxes` permet d'afficher le contour des boîtes ;
- `absolute` dispose les blocs relativement au coin supérieur gauche de la page (au lieu de la position courante) ;
- `overlay` dispose les blocs par-dessus le texte normal du document.

L'extension textpos

Quelques options de l'extension :

- `showboxes` permet d'afficher le contour des boîtes ;
- `absolute` dispose les blocs relativement au coin supérieur gauche de la page (au lieu de la position courante) ;
- `overlay` dispose les blocs par-dessus le texte normal du document.

Ceci devrait
apparaître
exactement
au centre de
la page

Code



Fichier Beamer3/code49.tex

```
\begin{textblock}{2}[0.5,0.5](8,8)  
%%On a utilisé les options absolute,  
%%showboxes et overlay  
%%Ceci devrait apparaître exactement  
%%au centre de la page  
\end{textblock}
```

Sortie

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
 - Manipulation de fichiers PostScript et PDF
 - Listings et programmes informatiques
 - Partitions de musique
 - Lettres et courrier
 - Affiches et posters
 - **Les extensions pstricks**
 - Transparents avec Seminar
 - Présentations avec Beamer
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Les extensions pstricks

Il s'agit d'un ensemble d'extensions pour réaliser des dessins dans des domaines très variés :

- `pstricks` pour les figures de base ;

Les extensions pstricks

Il s'agit d'un ensemble d'extensions pour réaliser des dessins dans des domaines très variés :

- `pstricks` pour les figures de base ;
- `pstricks-add` corrige certains bugs et ajoute des fonctionnalités supplémentaires (notation infix pour les formules entre autres) ;
- `pst-grad` pour colorier avec des gradients de couleur ;
- `pst-text` pour la manipulation du texte ;

Les extensions pstricks

Il s'agit d'un ensemble d'extensions pour réaliser des dessins dans des domaines très variés :

- `pstricks` pour les figures de base ;
- `pstricks-add` corrige certains bugs et ajoute des fonctionnalités supplémentaires (notation infix pour les formules entre autres) ;
- `pst-grad` pour colorier avec des gradients de couleur ;
- `pst-text` pour la manipulation du texte ;
- `pst-plot`, `pst-2dplot` et `pst-3dplot` pour des graphiques en 2D ou 3D, avec support Matlab ;

Les extensions pstricks

Il s'agit d'un ensemble d'extensions pour réaliser des dessins dans des domaines très variés :

- `pstricks` pour les figures de base ;
- `pstricks-add` corrige certains bugs et ajoute des fonctionnalités supplémentaires (notation infix pour les formules entre autres) ;
- `pst-grad` pour colorier avec des gradients de couleur ;
- `pst-text` pour la manipulation du texte ;
- `pst-plot`, `pst-2dplot` et `pst-3dplot` pour des graphiques en 2D ou 3D, avec support Matlab ;
- `pst-all` charge la plupart de ces extensions (mais ralentit aussi la compilation) ;

Les extensions pstricks

Il s'agit d'un ensemble d'extensions pour réaliser des dessins dans des domaines très variés :

- `pstricks` pour les figures de base ;
- `pstricks-add` corrige certains bugs et ajoute des fonctionnalités supplémentaires (notation infix pour les formules entre autres) ;
- `pst-grad` pour colorier avec des gradients de couleur ;
- `pst-text` pour la manipulation du texte ;
- `pst-plot`, `pst-2dplot` et `pst-3dplot` pour des graphiques en 2D ou 3D, avec support Matlab ;
- `pst-all` charge la plupart de ces extensions (mais ralentit aussi la compilation) ;
- `pst-tree` pour dessiner des arbres (syntaxiques ou autres) ;
- `pst-optic` pour dessiner des schémas optiques ;
- ...

Les extensions pstricks

Elles utilisent le langage bas niveau des fichiers PostScript, on ne peut donc pas utiliser Pdf \LaTeX et la plupart des visionneuses DVI ne les affichent pas.

Les extensions pstricks

Elles utilisent le langage bas niveau des fichiers PostScript, on ne peut donc pas utiliser Pdf \LaTeX et la plupart des visionneuses DVI ne les affichent pas.

Généralement on compile avec \LaTeX , puis on convertit avec `dvips` (et éventuellement `ps2pdf`).

Les extensions pstricks

Elles utilisent le langage bas niveau des fichiers PostScript, on ne peut donc pas utiliser Pdf \LaTeX et la plupart des visionneuses DVI ne les affichent pas.

Généralement on compile avec \LaTeX , puis on convertit avec `dvips` (et éventuellement `ps2pdf`).

Certains logiciels de dessin vectoriel permettent d'exporter du code PSTricks. Il est préférable d'utiliser `\input` ou `\include` depuis un fichier annexe à l'endroit désiré.

Les extensions pstricks

Elles utilisent le langage bas niveau des fichiers PostScript, on ne peut donc pas utiliser Pdf \LaTeX et la plupart des visionneuses DVI ne les affichent pas.

Généralement on compile avec \LaTeX , puis on convertit avec `dvips` (et éventuellement `ps2pdf`).

Certains logiciels de dessin vectoriel permettent d'exporter du code PSTricks. Il est préférable d'utiliser `\input` ou `\include` depuis un fichier annexe à l'endroit désiré.

Si on souhaite malgré tout utiliser Pdf \LaTeX , il est possible de compiler séparément les dessins PSTricks, les convertir en PDF puis les inclure avec `\includegraphics` : c'est ce que fait l'extension `pdftricks`.

L'extension pdftricks

Définit les environnements `psinputs` pour les parties du préambule spécifiques à PSTricks et `pdfpic` pour les dessins dans le document.

L'extension pdftricks

Définit les environnements `psinputs` pour les parties du préambule spécifiques à PSTricks et `pdfpic` pour les dessins dans le document.

Structure générale



Fichier Beamer3/code52.tex

```
\documentclass{article}

\usepackage{pdftricks}

\begin{psinputs}
  \usepackage{pstricks,pst-3d}
\end{psinputs}

\begin{document}

\begin{pdfpic}
  \begin{pspicture}(0,0)(10,10)
    ...
  \end{pspicture}
\end{pdfpic}

\end{document}
```

L'extension pdftricks

Définit les environnements `psinputs` pour les parties du préambule spécifiques à PSTricks et `pdfpic` pour les dessins dans le document.

Structure générale



Fichier Beamer3/code53.tex

```
\documentclass{article}

\usepackage{pdftricks}

\begin{psinputs}
  \usepackage{pstricks,pst-3d}
\end{psinputs}

\begin{document}

\begin{pdfpic}
  \begin{pspicture}(0,0)(10,10)
    ....
  \end{pspicture}
\end{pdfpic}

\end{document}
```

Il faudra rajouter l'option `-shell-escape` (ou `--enable-write18` sous Windows/MiKTeX) sur la ligne de commande pour autoriser \LaTeX à lancer les compilations externes nécessaires.

Primitives

Par défaut, les primitives de dessin PSTricks n'occupent aucune place, et sont affichées sur la ligne de base à l'emplacement du prochain caractère.

Primitives

Par défaut, les primitives de dessin PSTricks n'occupent aucune place, et sont affichées sur la ligne de base à l'emplacement du prochain caractère.

Code



Fichier Beamer3/code55.tex

```
\ttfamily
%Le mot barré va être affiché par-
  dessus le segment
Un mot \psline(0,0)(2.5em,1ex) barré
  en diagonale.
```

Sortie

Un mot barré en diagonale.

Dessins PostScript

L'environnement `pspicture` permet de créer des assemblages de primitives dans une boîte de taille fixée.

Dessins PostScript

L'environnement `pspicture` permet de créer des assemblages de primitives dans une boîte de taille fixée.

`\begin{pspicture}(x0,y0)(x1,y1)` pour une boîte de taille $(x1 - x0) \times (y1 - y0)$.

Dessins PostScript

L'environnement `pspicture` permet de créer des assemblages de primitives dans une boîte de taille fixée.

`\begin{pspicture}(x0,y0)(x1,y1)` pour une boîte de taille $(x1 - x0) \times (y1 - y0)$.

La boîte sera disposée à la place du prochain caractère, de façon à ce que l'origine $(x0, y0)$ soit sur la ligne de base du texte.

Dessins PostScript

L'environnement `pspicture` permet de créer des assemblages de primitives dans une boîte de taille fixée.

`\begin{pspicture}(x0,y0)(x1,y1)` pour une boîte de taille $(x1 - x0) \times (y1 - y0)$.

La boîte sera disposée à la place du prochain caractère, de façon à ce que l'origine $(x0, y0)$ soit sur la ligne de base du texte.

Si l'on ne les précise pas, les unités par défaut sont le centimètre pour les longueurs et le degré pour les angles.

Dessins PostScript

L'environnement `pspicture` permet de créer des assemblages de primitives dans une boîte de taille fixée.

`\begin{pspicture}(x0,y0)(x1,y1)` pour une boîte de taille $(x1 - x0) \times (y1 - y0)$.

La boîte sera disposée à la place du prochain caractère, de façon à ce que l'origine $(x0, y0)$ soit sur la ligne de base du texte.

Si l'on ne les précise pas, les unités par défaut sont le centimètre pour les longueurs et le degré pour les angles.

Si l'on ne précise pas le premier argument $(x0, y0)$ alors ce sera $(0, 0)$.

Dessins PostScript

L'environnement `pspicture` permet de créer des assemblages de primitives dans une boîte de taille fixée.

`\begin{pspicture}(x0,y0)(x1,y1)` pour une boîte de taille $(x1 - x0) \times (y1 - y0)$.

La boîte sera disposée à la place du prochain caractère, de façon à ce que l'origine $(x0,y0)$ soit sur la ligne de base du texte.

Si l'on ne les précise pas, les unités par défaut sont le centimètre pour les longueurs et le degré pour les angles.

Si l'on ne précise pas le premier argument $(x0,y0)$ alors ce sera $(0,0)$.

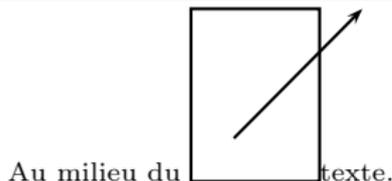
Code



Fichier Beamer3/code61.tex

```
Au milieu du
\begin{pspicture}(-1,-1)(2,3)
\psframe(-1,-1)(2,3)
%Cette flèche va dépasser à droite
de 1cm
\psline{->}(0,0)(3,3)
\end{pspicture}%
texte.
```

Sortie



Options globales

À tout endroit du document on peut changer certains paramètres avec
`\psset{param1=value1,param2=value2,...}`

Options globales

À tout endroit du document on peut changer certains paramètres avec `\psset{param1=value1,param2=value2,...}`, en particulier :

- `unit = <longueur>` pour changer l'unité de mesure par défaut (qui vaut 1cm par défaut) ;

Options globales

À tout endroit du document on peut changer certains paramètres avec `\psset{param1=value1,param2=value2,...}`, en particulier :

- `unit = <longueur>` pour changer l'unité de mesure par défaut (qui vaut 1cm par défaut) ;
- `origin = {\langle x \rangle , \langle y \rangle}` pour déplacer l'origine (`{0,0}` par défaut) ;

Options globales

À tout endroit du document on peut changer certains paramètres avec `\psset{param1=value1,param2=value2,...}`, en particulier :

- `unit = <longueur>` pour changer l'unité de mesure par défaut (qui vaut 1cm par défaut) ;
- `origin = {<x> , <y>}` pour déplacer l'origine (`{0,0}` par défaut) ;
- `algebraic = <true,false>` pour activer ou non la notation infixe des formules (par défaut en notation polonaise). Nécessite l'extension `pstricks-add`.

Options globales

À tout endroit du document on peut changer certains paramètres avec `\psset{param1=value1,param2=value2,...}`, en particulier :

- `unit = <longueur>` pour changer l'unité de mesure par défaut (qui vaut 1cm par défaut) ;
- `origin = {<x>, <y>}` pour déplacer l'origine ($\{0,0\}$ par défaut) ;
- `algebraic = <true,false>` pour activer ou non la notation infixe des formules (par défaut en notation polonaise). Nécessite l'extension `pstricks-add`.

La portée des changements est limitée au groupe en cours.

Options globales

À tout endroit du document on peut changer certains paramètres avec `\psset{param1=value1,param2=value2,...}`, en particulier :

- `unit = <longueur>` pour changer l'unité de mesure par défaut (qui vaut 1cm par défaut) ;
- `origin = {\langle x \rangle, \langle y \rangle}` pour déplacer l'origine ($\{0,0\}$ par défaut) ;
- `algebraic = <true,false>` pour activer ou non la notation infixe des formules (par défaut en notation polonaise). Nécessite l'extension `pstricks-add`.

La portée des changements est limitée au groupe en cours.

`\newsstyle{nom}{paramètres}` permet de définir des ensembles de paramètres qu'on peut activer facilement avec `\psset{style = <nom>}`.

Commandes personnalisées

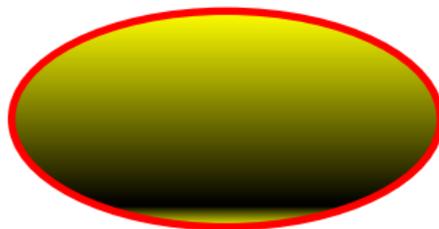
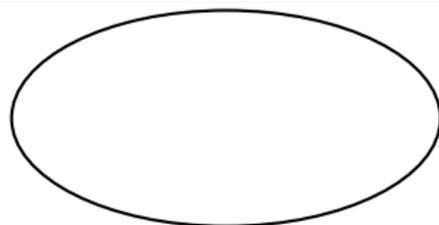
Code



Fichier Beamer3/code62.tex

```
\newpsobject{superpsellipse}{  
  psellipse}{linecolor=red,  
  linewidth=2pt,fillstyle=gradient,  
  gradbegin=yellow,gradend=black}  
\begin{pspicture}(0,0)(5,5)  
  %%Ellipse normale  
  \psellipse(2.5,4)(2,1)  
  %%Ellipse personnalisée  
  \superpsellipse(2.5,1)(2,1)  
\end{pspicture}
```

Sortie



Couleurs

- `\newgray{nom}{valeur}` permet de définir un nouveau niveau de gris (qui peut ensuite être activé n'importe où avec `\nom`).

Couleurs

- `\newgray{nom}{valeur}` permet de définir un nouveau niveau de gris (qui peut ensuite être activé n'importe où avec `\nom`). Les niveaux suivants sont prédéfinis :

`black` `darkgray` `gray` `lightgray` `white`

Couleurs

- `\newgray{nom}{valeur}` permet de définir un nouveau niveau de gris (qui peut ensuite être activé n'importe où avec `\nom`). Les niveaux suivants sont prédéfinis :

`black` `darkgray` `gray` `lightgray` `white`

- `\newrgbcolor{nom}{R G B}` permet de définir une nouvelle couleur RGB.

Couleurs

- `\newgray{nom}{valeur}` permet de définir un nouveau niveau de gris (qui peut ensuite être activé n'importe où avec `\nom`). Les niveaux suivants sont prédéfinis :

`black` `darkgray` `gray` `lightgray` `white`

- `\newrgbcolor{nom}{R G B}` permet de définir une nouvelle couleur RGB. Les couleurs suivantes sont prédéfinies :

`red` `green` `blue` `cyan` `magenta` `yellow` `brown`
`lime` `olive` `orange` `pink` `purple` `teal` `violet`

Couleurs

- `\newgray{nom}{valeur}` permet de définir un nouveau niveau de gris (qui peut ensuite être activé n'importe où avec `\nom`). Les niveaux suivants sont prédéfinis :

`black` `darkgray` `gray` `lightgray` `white`

- `\newrgbcolor{nom}{R G B}` permet de définir une nouvelle couleur RGB. Les couleurs suivantes sont prédéfinies :

`red` `green` `blue` `cyan` `magenta` `yellow` `brown`
`lime` `olive` `orange` `pink` `purple` `teal` `violet`

Code



Fichier Beamer3/code67.tex

```
\newrgbcolor{CHAMEAU}{0.9_0.4_0.1}  
{\CHAMEAU_couleur}_désactivée  
\psframe*[linecolor=CHAMEAU](0,0)(1,1  
ex)
```

Sortie

couleur désactivée 

Systèmes de coordonnées

La commande `\SpecialCoor` permet d'utiliser des systèmes de coordonnées supplémentaires

Systèmes de coordonnées

La commande `\SpecialCoor` permet d'utiliser des systèmes de coordonnées supplémentaires :

- les coordonnées cartésiennes usuelles (x, y) disponibles par défaut ;

Systemes de coordonnees

La commande `\SpecialCoor` permet d'utiliser des systemes coordonnees supplementaires :

- les coordonnees cartesiennes usuelles (x,y) disponibles par defaut ;
- les coordonnees polaires $(\text{rayon};\text{angle})$;

Systemes de coordonnees

La commande `\SpecialCoor` permet d'utiliser des systemes coordonnees supplementaires :

- les coordonnees cartesiennes usuelles (x,y) disponibles par defaut ;
- les coordonnees polaires $(\text{rayon};\text{angle})$;
- etc. . .

Systèmes de coordonnées

La commande `\SpecialCoor` permet d'utiliser des systèmes de coordonnées supplémentaires :

- les coordonnées cartésiennes usuelles (x, y) disponibles par défaut ;
- les coordonnées polaires $(\text{rayon}; \text{angle})$;
- etc. . .

La commande `\degrees[valeur]` permet de travailler avec une unité d'angle correspondant à $\frac{1}{\text{valeur}}$ du cercle trigonométrique.

Systèmes de coordonnées

La commande `\SpecialCoor` permet d'utiliser des systèmes de coordonnées supplémentaires :

- les coordonnées cartésiennes usuelles (x,y) disponibles par défaut ;
- les coordonnées polaires $(\text{rayon};\text{angle})$;
- etc. . .

La commande `\degrees[valeur]` permet de travailler avec une unité d'angle correspondant à $\frac{1}{\text{va$ leur} du cercle trigonométrique.

Code

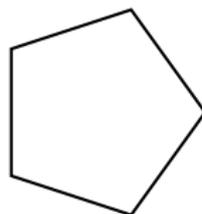


Fichier Beamer3/code73.tex

```
\SpecialCoor
\degrees[5]

\begin{pspicture}(-2.5,-1)(2.5,1)
%%Dessin facile d'un pentagone
  régulier
%%\pspolygon(1;0)(1;1)(1;2)(1;3)(1;4)
\end{pspicture}
```

Sortie



Points

```
\psdots[options](x1,y1)(x2,y2)\dots(xn,yn).
```

Points

`\psdots[options](x1,y1)(x2,y2)...`(xn,yn).

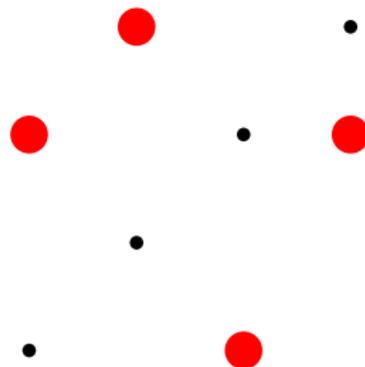
Code



Fichier Beamer3/code75.tex

```
\begin{pspicture}(0,0)(5,5)
\psdots(1,1)(2,2)(3,3)(4,4)
\psdots[linecolor=red,dotsize=10pt]
(2,4)(3,1)(4,3)(1,3)
\end{pspicture}
```

Sortie



Options spécifiques des points

- `dotstyle = <o, *, +, x, triangle, triangle*, square, square*, diamond, diamond*, pentagon, pentagon*, ...>`

pour obtenir respectivement



Options spécifiques des points

- `dotstyle = <o, *, +, x, triangle, triangle*, square, square*, diamond, diamond*, pentagon, pentagon*, ...>`

pour obtenir respectivement



- `dotscale = <valeur>` pour augmenter la taille par défaut des points.

Segments et lignes brisées

```
\psline[options]{extrémités}(x1,y1)(x2,y2)...(xn,yn).
```

Segments et lignes brisées

`\psline[options]{extrémités}(x1,y1)(x2,y2)...(xn,yn).`

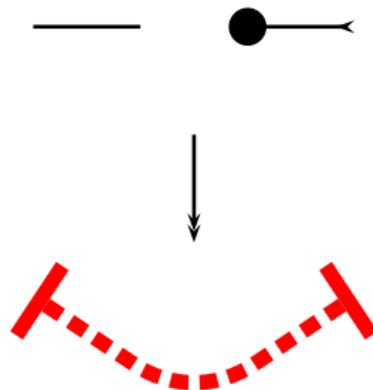
Code



Fichier Beamer3/code77.tex

```
\begin{pspicture}(0,0)(5,5)
uu\psline(1,4)(2,4)
uu\psline{*-<} (3,4)(4,4)
uu\psline{->>} (2.5,3)(2.5,2)
uu\psline[linearc=1,linestyle=dashed,
linecolor=red,linewidth=4pt
]{|-|}(1,1.5)(2.5,0.5)(4,1.5)
\end{pspicture}
```

Sortie



Options spécifiques des lignes

- `linewidth = <longueur>` pour la largeur de trait ;
- `linecolor = <couleur>` pour la couleur des traits (ou du remplissage pour les commandes étoilées) ;
- `linestyle = <none, solid, dashed, dotted>` pour le type de trait :



- `doubleline = <true, false>` pour doubler le trait ;

Options spécifiques des lignes

- `linewidth = <longueur>` pour la largeur de trait ;
- `linecolor = <couleur>` pour la couleur des traits (ou du remplissage pour les commandes étoilées) ;
- `linestyle = <none, solid, dashed, dotted>` pour le type de trait :



- `doubleline = <true, false>` pour doubler le trait ;
- `showpoints = <true, false>` pour afficher ou non les extrémités des éléments ;

Options spécifiques des lignes

- `linewidth = <longueur>` pour la largeur de trait ;
- `linecolor = <couleur>` pour la couleur des traits (ou du remplissage pour les commandes étoilées) ;
- `linestyle = <none, solid, dashed, dotted>` pour le type de trait :



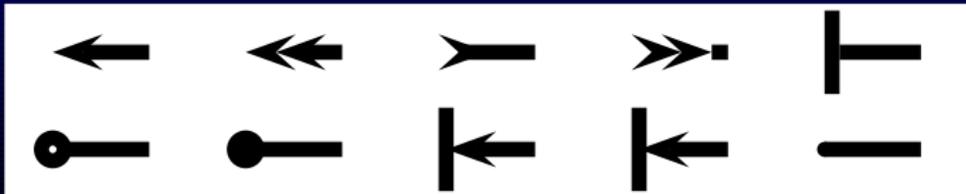
- `doubleline = <true, false>` pour doubler le trait ;
- `showpoints = <true, false>` pour afficher ou non les extrémités des éléments ;
- `linearc = <longueur>` pour le rayon de courbure utilisé pour arrondir les angles ;

Options spécifiques des lignes

- `linewidth` = $\langle \text{longueur} \rangle$ pour la largeur de trait ;
- `linecolor` = $\langle \text{couleur} \rangle$ pour la couleur des traits (ou du remplissage pour les commandes étoilées) ;
- `linestyle` = $\langle \text{none, solid, dashed, dotted} \rangle$ pour le type de trait :



- `doubleline` = $\langle \text{true, false} \rangle$ pour doubler le trait ;
- `showpoints` = $\langle \text{true, false} \rangle$ pour afficher ou non les extrémités des éléments ;
- `linearc` = $\langle \text{longueur} \rangle$ pour le rayon de courbure utilisé pour arrondir les angles ;
- les extrémités sont de la forme $\{ \dots - \dots \}$ avec les possibilités $\langle, \ll, \rangle, \gg, |, o, *, | \langle, \rangle |$ ou `c` :



Rectangles

`\psframe[options](x1,y1)(x2,y2)`, ou la variante étoilée `\psframe*` pour un rectangle plein.

Rectangles

`\psframe[options](x1,y1)(x2,y2)`, ou la variante étoilée `\psframe*` pour un rectangle plein.

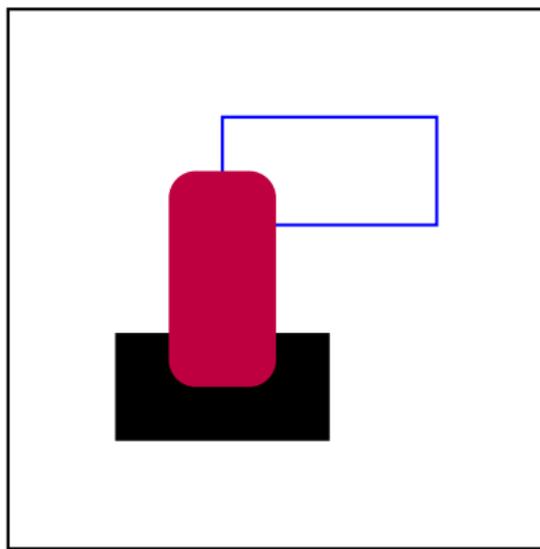
Code



Fichier Beamer3/code79.tex

```
\begin{pspicture}(0,0)(5,5)
  \psframe(0,0)(5,5)
  \psframe*(1,1)(3,2)
  \psframe[fillcolor=red,linecolor=
    blue](2,3)(4,4)
  \psframe*[framearc=0.5,linecolor=
    purple,fillstyle=solid](1.5,1.5)
    (2.5,3.5)
\end{pspicture}
```

Sortie



Polygones

On utilise `\pspolygon[options](x1,y1)(x2,y2)\dots(xn,yn)`, ou la variante étoilée `\pspolygon*` pour un polygone plein.

Polygones

On utilise `\pspolygon[options](x1,y1)(x2,y2)...`(x_n,y_n), ou la variante étoilée `\pspolygon*` pour un polygone plein.

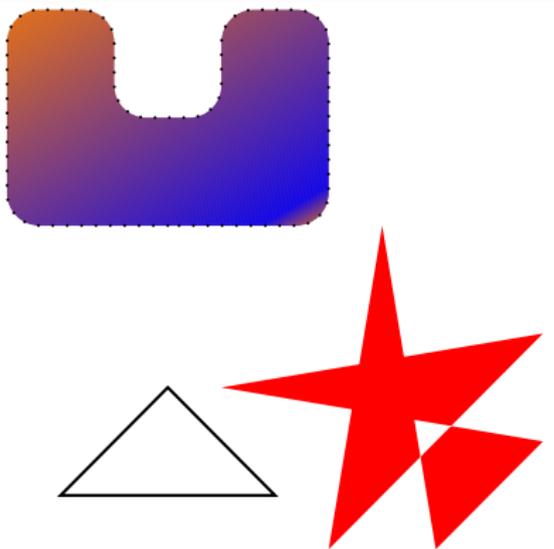
Code



Fichier Beamer3/code81.tex

```
\begin{pspicture}(0,0)(5,5)
\pspolygon(0.5,0.5)(1.5,1.5)
(2.5,0.5)
\pspolygon[linearc=0.3,linestyle=
dotted,fillstyle=gradient,
gradbegin=orange,gradend=blue,
gradangle=30](0,3)(0,5)(1,5)(1,4)
(2,4)(2,5)(3,5)(3,3)
\pspolygon*[linecolor=red](3,0)
(3.5,3)(4,0)(5,1)(2,1.5)(5,2)
\end{pspicture}
```

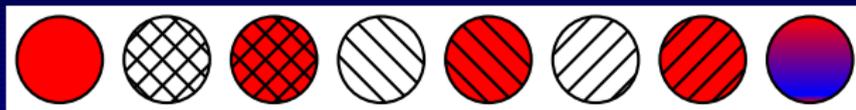
Sortie



Options spécifiques de remplissage

- `fillcolor = <couleur>` pour la couleur de remplissage ;
- `fillstyle = <solid, crosshatch, crosshatch*, vlines, vlines*, pour
hlines, hlines*, gradient>`

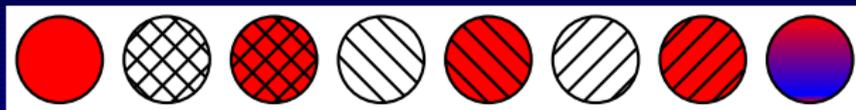
le motif de remplissage :



Options spécifiques de remplissage

- `fillcolor = <couleur>` pour la couleur de remplissage ;
- `fillstyle = <solid, crosshatch, crosshatch*, vlines, vlines*, pour hlines, hlines*, gradient>`

le motif de remplissage :

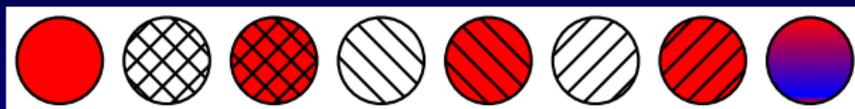


- `hatchcolor = <couleur>` et `hatchwidth = <longueur>` pour la couleur et la largeur de trait des hachures ;
- `hatchsep = <longueur>` et `hatchangle = <angle>` pour l'espacement et l'inclinaison des hachures ;

Options spécifiques de remplissage

- `fillcolor = <couleur>` pour la couleur de remplissage ;
- `fillstyle = <solid, crosshatch, crosshatch*, vlines, vlines*, pour hlines, hlines*, gradient>`

le motif de remplissage :

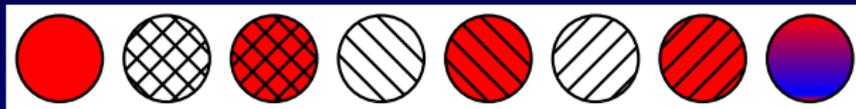


- `hatchcolor = <couleur>` et `hatchwidth = <longueur>` pour la couleur et la largeur de trait des hachures ;
- `hatchsep = <longueur>` et `hatchangle = <angle>` pour l'espacement et l'inclinaison des hachures ;
- `gradbegin = <couleur>` et `gradend = <couleur>` pour les couleurs de dégradé (nécessite `pst-grad`) ;
- `gradmidpoint = <valeur>` pour décaler le dégradé ;

Options spécifiques de remplissage

- `fillcolor` = $\langle \text{couleur} \rangle$ pour la couleur de remplissage ;
- `fillstyle` = $\langle \text{solid, crosshatch, crosshatch*, vlines, vlines*, pour hlines, hlines*, gradient} \rangle$

le motif de remplissage :



- `hatchcolor` = $\langle \text{couleur} \rangle$ et `hatchwidth` = $\langle \text{longueur} \rangle$ pour la couleur et la largeur de trait des hachures ;
- `hatchsep` = $\langle \text{longueur} \rangle$ et `hatchangle` = $\langle \text{angle} \rangle$ pour l'espacement et l'inclinaison des hachures ;
- `gradbegin` = $\langle \text{couleur} \rangle$ et `gradend` = $\langle \text{couleur} \rangle$ pour les couleurs de dégradé (nécessite `pst-grad`) ;
- `gradmidpoint` = $\langle \text{valeur} \rangle$ pour décaler le dégradé ;
- `framearc` = $\langle \text{longueur} \rangle$ pour le rayon de courbure utilisé pour arrondir les angles.

Cercles, ellipses et arcs

- `\pscircle[options](x,y){r}` pour un cercle (ou `\pscircle*` pour un disque);

Cercles, ellipses et arcs

- `\pscircle[options](x,y){r}` pour un cercle (ou `\pscircle*` pour un disque);
- `\psellipse[options](x,y)(rayon horizontal,rayon vertical)` pour une ellipse (ou `\psellipse*` pour un disque elliptique);

Cercles, ellipses et arcs

- `\pscircle[options](x,y){r}` pour un cercle (ou `\pscircle*` pour un disque);
- `\psellipse[options](x,y)(rayon horizontal,rayon vertical)` pour une ellipse (ou `\psellipse*` pour un disque elliptique);
- `\psarc[options]{extrémités}(x,y){r}{angle1}{angle2}` pour l'arc de cercle limité par les angles (ou `\psarc*` pour remplir la région délimitée par la corde);

Cercles, ellipses et arcs

- `\pscircle[options](x,y){r}` pour un cercle (ou `\pscircle*` pour un disque);
- `\psellipse[options](x,y)(rayon horizontal,rayon vertical)` pour une ellipse (ou `\psellipse*` pour un disque elliptique);
- `\psarc[options]{extrémités}(x,y){r}{angle1}{angle2}` pour l'arc de cercle limité par les angles (ou `\psarc*` pour remplir la région délimitée par la corde);
- `\pswedge[options](x,y){r}{angle1}{angle2}` pour une part de camembert (`\pswedge*` pour la remplir).

Cercles, ellipses et arcs

- `\pscircle[options](x,y){r}` pour un cercle (ou `\pscircle*` pour un disque);
- `\psellipse[options](x,y)(rayon horizontal,rayon vertical)` pour une ellipse (ou `\psellipse*` pour un disque elliptique);
- `\psarc[options]{extrémités}(x,y){r}{angle1}{angle2}` pour l'arc de cercle limité par les angles (ou `\psarc*` pour remplir la région délimitée par la corde);
- `\pswedge[options](x,y){r}{angle1}{angle2}` pour une part de camembert (`\pswedge*` pour la remplir).

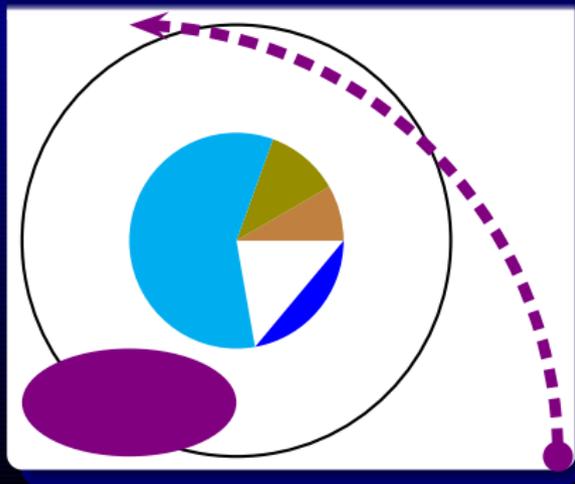
Code



Fichier Beamer3/code86.tex

```
\begin{pspicture}(0,0)(5,4)
uu\pscircle(2,2){2}
uu\psellipse*[linecolor=violet
](1,0.5)(1,0.5)
uu\psarc[linecolor=violet,linewidth=3
pt,linestyle=dashed]{*->}(1,0)
{4}{0}{90}
uu\pswedge*[linecolor=brown](2,2)
{1}{0}{30}
uu\pswedge*[linecolor=olive](2,2)
{1}{30}{70}
uu\pswedge*[linecolor=cyan](2,2)
{1}{70}{280}
uu\psarc*[linecolor=blue](2,2)
{1}{280}{360}
\end{pspicture}
```

Sortie



Courbes

Les variantes étoilées ferment la courbe avec une corde et remplissent la région.

- `\pscurve[options](x1,y1)(x2,y2) \dots (xn,yn)` pour tracer une courbe d'interpolation passant par tous les points (`\psccurve` ferme la courbe);

Courbes

Les variantes étoilées ferment la courbe avec une corde et remplissent la région.

- `\pscurve[options](x1,y1)(x2,y2) \dots (xn,yn)` pour tracer une courbe d'interpolation passant par tous les points (`\psccurve` ferme la courbe);
- `\psecurve` passe par tous les points sauf ceux des extrémités (qui servent à contrôler les tangentes);

Courbes

Les variantes étoilées ferment la courbe avec une corde et remplissent la région.

- `\pscurve[options](x1,y1)(x2,y2) . . . (xn,yn)` pour tracer une courbe d'interpolation passant par tous les points (`\psccurve` ferme la courbe);
- `\psecurve` passe par tous les points sauf ceux des extrémités (qui servent à contrôler les tangentes);
- `\psbezier` trace une courbe de Bézier.

Courbes

Les variantes étoilées ferment la courbe avec une corde et remplissent la région.

- `\pscurve[options](x1,y1)(x2,y2) \dots (xn,yn)` pour tracer une courbe d'interpolation passant par tous les points (`\psccurve` ferme la courbe);
- `\psecurve` passe par tous les points sauf ceux des extrémités (qui servent à contrôler les tangentes);
- `\psbezier` trace une courbe de Bézier.

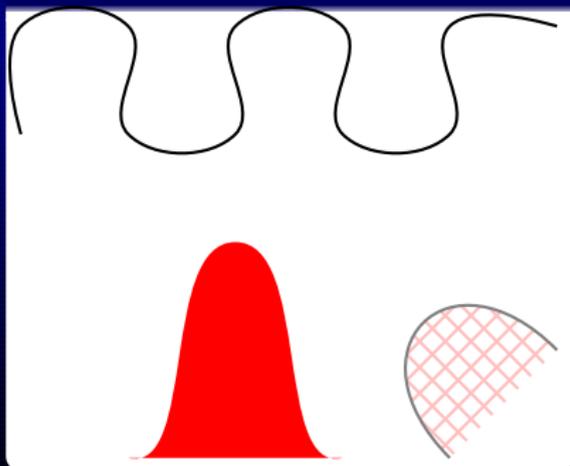
Code



Fichier Beamer3/code90.tex

```
\begin{pspicture}(0,0)(5,4)
\pscurve(0,3)(0,4)(1,4)(1,3)(2,3)
(2,4)(3,4)(3,3)(4,3)(4,4)(5,4)
\psecurve*[linecolor=red](1,1)(1,0)
(2,2)(3,0)(3,1)
\psbezier[linecolor=gray,fillstyle=
crosshatch,hatchcolor=pink](4,0)
(3,1)(4,2)(5,1)
\end{pspicture}
```

Sortie



Translations et rotations

- `\rput[référence]{rotation}(x,y){texte}` pour placer du texte relativement au point (x,y) . La référence (par défaut centrée) peut être une combinaison de B (ligne de base), b (bas) ou t (haut) avec l (gauche) ou r (droite);

Translations et rotations

- `\rput[référence]{rotation}(x,y){texte}` pour placer du texte relativement au point (x,y) . La référence (par défaut centrée) peut être une combinaison de B (ligne de base), b (bas) ou t (haut) avec l (gauche) ou r (droite);
- `\uput{distance}[direction]{rotation}(x,y){texte}` pour placer du texte à distance donnée dans une direction.

Translations et rotations

- `\rput[référence]{rotation}(x,y){texte}` pour placer du texte relativement au point (x,y) . La référence (par défaut centrée) peut être une combinaison de B (ligne de base), b (bas) ou t (haut) avec l (gauche) ou r (droite);
- `\uput{distance}[direction]{rotation}(x,y){texte}` pour placer du texte à distance donnée dans une direction.

Code



Fichier Beamer3/code93.tex

```
\begin{pspicture}(0,0)(5,4)
  \psdots[linecolor=red](1,4)
  \rput[t](1,4){\ttfamily\
    textbackslash\rput(1,4)}
  \rput{45}(2.5,2){Au centre}
  \uput{5}[30]{-90}(0,0){\huge\
    \ttfamily\textbackslash\uput}
  \uput{4}[-70](0,4){\color{orange}\
    \huge\ttfamily\textbackslash\uput}
\end{pspicture}
```

Sortie

`\rput`(1,4)

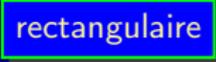
Au centre

`\uput`

`\uput`

Cadres

Ces commandes tracent un cadre autour de leur argument et les variantes étoilées remplissent le cadre :

- `\psframebox{texte}` pour un cadre  (équivalent de `\fbox`);
- `\psdblframebox{texte}` pour un cadre  à filet double;
- `\psshadowbox{\white texte}` pour un cadre  ombré;
- `\pscircularbox{texte}` pour un cadre  ;
- `\psovalbox{texte}` pour un cadre  ;
- `\psdiabox{texte}` pour un cadre  ;
- `\pstribox{texte}` pour un cadre  .

Quadrillage

`\psgrid[options](x0,y0)(xmin,ymin)(xmax,ymax)` affiche un quadrillage sur le pavé $[xmin, ymin] \times [xmax, ymax]$. L'unité des graduations est $x0-xmin$ en abscisses et $y0-ymin$ en ordonnées (par défaut 1).

Quadrillage

`\psgrid[options](x0,y0)(xmin,ymin)(xmax,ymax)` affiche un quadrillage sur le pavé $[xmin, ymin] \times [xmax, ymax]$. L'unité des graduations est $x0-xmin$ en abscisses et $y0-ymin$ en ordonnées (par défaut 1).

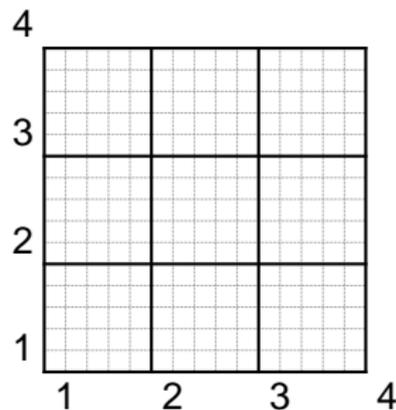
Code



Fichier Beamer3/code95.tex

```
\begin{pspicture}(0,0)(5,5)
\psgrid[subgriddots=10,subgridcolor
=gray](1,1)(4,4)
\end{pspicture}
```

Sortie



Options spécifiques des quadrillages

- `subgriddiv = <n>` pour afficher `n` sous-graduations entre chaque graduation principale ;
- `griddots = <n>` et `subgriddots = <n>` pour remplacer les traits pleins par `n` points entre chaque graduation ;
- `gridcolor = <couleur>` et `subgridcolor = <couleur>` pour la couleur des graduations ;
- `gridwidth = <longueur>` et `subgridwidth = <longueur>` pour l'épaisseur des traits de graduation ;
- `gridlabels = <longueur>` pour la taille de fontes des étiquettes.

Repères et axes

`\psaxes[options]{extrémités}(x0,y0)(xmin,ymin)(xmax,ymax)` affiche deux axes perpendiculaires dans le pavé $[xmin, ymin] \times [xmax, ymax]$ et se coupant en $(x0, y0)$.

Repères et axes

`\psaxes[options]{extrémités}(x0,y0)(xmin,ymin)(xmax,ymax)` affiche deux axes perpendiculaires dans le pavé $[xmin, ymin] \times [xmax, ymax]$ et se coupant en $(x0, y0)$.

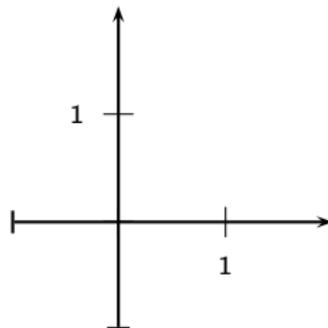
Code



Fichier Beamer3/code97.tex

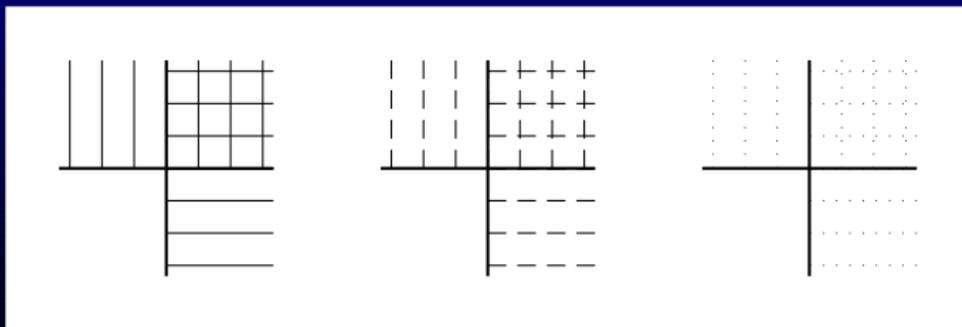
```
\begin{pspicture}(0,0)(5,5)  
  \psaxes{|->}(2,2)(1,1)(4,4)  
\end{pspicture}
```

Sortie



Options spécifiques des axes

- $Dx = \langle \text{longueur} \rangle$ et $Dy = \langle \text{longueur} \rangle$ pour l'espacement entre les graduations ;
- `comma` pour utiliser la virgule comme séparateur décimal ;
- `labels = \langle xy, x, y, none \rangle` pour contrôler l'affichage des étiquettes ;
- `ticks = \langle xy, x, y, none \rangle` pour contrôler l'affichage des graduations ;
- `xAxis = \langle true, false \rangle` et `yAxis = \langle true, false \rangle` pour activer ou désactiver un axe ;
- `ticksize = \langle longueur \rangle` pour la taille des graduations ;
- `ticklinestyle = \langle solid, dashed, dotted \rangle` pour le style des graduations :



Tracé de fonctions

On utilise `\psplot[options]{xmin}{xmax}{fonction}`.

Tracé de fonctions

On utilise `\psplot[options]{xmin}{xmax}{fonction}`.

Par défaut les fonctions sont données en notation polonaise, l'option `algebraic` permet de passer en notation infixe.

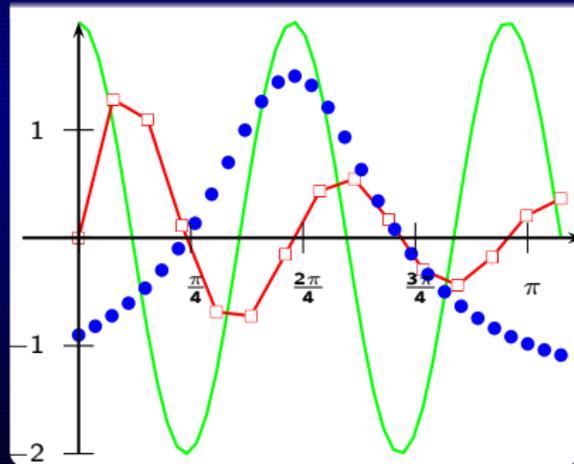
Code



Fichier Beamer3/code99.tex

```
\begin{pspicture}(-0.5,-2)(4.5,2)
\psplot[linecolor=green]{0}{4.5}{x
  180mulcos2mul}
\psplot[algebraic,linecolor=red,
  plotpoints=15,showpoints=true,
  dotstyle=square]{0}{4.5}{2*sin
  (3.14159*x)/(x+1)}
\psplot[algebraic,plotpoints=30,
  plotstyle=dots,dotstyle=*,
  linecolor=blue]{0}{4.5}{3/((x-2)
  ^2+1)-1.5}
\psaxes[trigLabels=true,
  trigLabelBase=4,xunit=\pstRadUnit]
{->}(0,0)(-0.5,-2)(4.5,2)
\end{pspicture}
```

Sortie



Plots

On utilise `\savedata\commande [data]` pour stocker des coordonnées dans une commande.

Plots

On utilise `\savedata\commande [data]` pour stocker des coordonnées dans une commande.

On peut ensuite utiliser `\dataplot [options]{\commande}` pour l'affichage.

Plots

On utilise `\savedata\commande[data]` pour stocker des coordonnées dans une commande.

On peut ensuite utiliser `\dataplot[options]{\commande}` pour l'affichage.

L'option `plotstyle` peut prendre les valeurs `dots`, `line`, `polygon`, `curve`, `ecurve` ou `ccurve`.

Plots

On utilise `\savedata\commande[data]` pour stocker des coordonnées dans une commande.

On peut ensuite utiliser `\dataplot[options]{\commande}` pour l'affichage.

L'option `plotstyle` peut prendre les valeurs `dots`, `line`, `polygon`, `curve`, `ecurve` ou `ccurve`.

Code

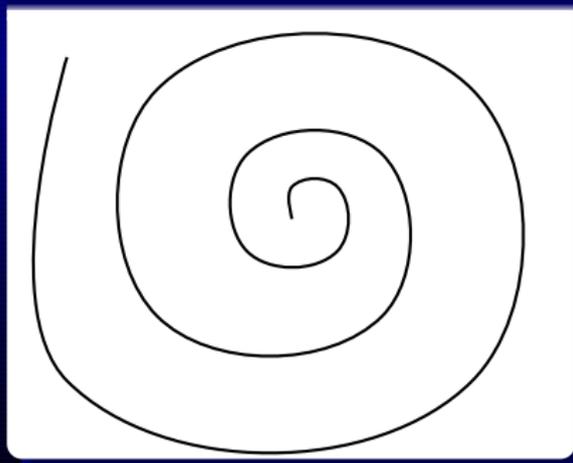


Fichier Beamer3/code103.tex

```
\savedata\MYDATA [0_0_0_1_1_1_1_1_-1_-1_
-1_-1_2_2_2_2_-3_-3_-3_4_4_4_4
_5_5_5_5_5]

\begin{pspicture}(-6,-7)(6,6)
\dataplot[plotstyle=curve]{\MYDATA}
\end{pspicture}
```

Sortie



Importation de données externes

Les données peuvent être générées par Matlab ou Mathematica par exemple.

Importation de données externes

Les données peuvent être générées par Matlab ou Mathematica par exemple.

Elles doivent être constituées de couples de coordonnées séparés par d'autres caractères (espaces, virgules, etc. . .).

Les autres caractères (parenthèses, crochets, etc. . .) sont ignorés.

Importation de données externes

Les données peuvent être générées par Matlab ou Mathematica par exemple.

Elles doivent être constituées de couples de coordonnées séparés par d'autres caractères (espaces, virgules, etc...).

Les autres caractères (parenthèses, crochets, etc...) sont ignorés.

- `\readdata\commande{fichier}` définit une commande pour stocker les données du fichier (pour utilisation avec `\dataplot`);

Importation de données externes

Les données peuvent être générées par Matlab ou Mathematica par exemple.

Elles doivent être constituées de couples de coordonnées séparés par d'autres caractères (espaces, virgules, etc...).

Les autres caractères (parenthèses, crochets, etc...) sont ignorés.

- `\readdata\commande{fichier}` définit une commande pour stocker les données du fichier (pour utilisation avec `\dataplot`);
- `\fileplot[options]{fichier}` affiche directement les données d'un fichier.

Importation de données externes

Les données peuvent être générées par Matlab ou Mathematica par exemple.

Elles doivent être constituées de couples de coordonnées séparés par d'autres caractères (espaces, virgules, etc...).

Les autres caractères (parenthèses, crochets, etc...) sont ignorés.

- `\readdata\commande{fichier}` définit une commande pour stocker les données du fichier (pour utilisation avec `\dataplot`);
- `\fileplot[options]{fichier}` affiche directement les données d'un fichier.

Code



Fichier Beamer3/code108.tex

```
\readdata\COEUR{Coeur.txt}

\begin{pspicture}(-2.5,-1)(2.5,2)
\fileplot*[linecolor=red]{Coeur.txt}
\end{pspicture}

\dataplot[linecolor=pink,unit=1.5cm]
{\COEUR}
\end{pspicture}
```

Sortie



Régions

```
\pscustom[options]{frontière}.
```

Régions

`\pscustom[options]{frontière}`.

Si la frontière spécifiée n'est pas connexe, les extrémités sont reliées par des cordes pour définir la zone de remplissage.

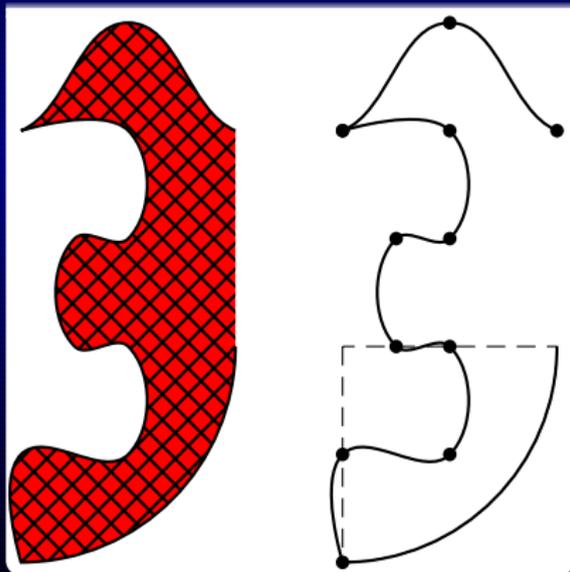
Code



Fichier Beamer3/code110.tex

```
\begin{pspicture}(0,0)(5,5)
uu\newcommand\perimeter{
uuuu\psecurve(3,4)(2,4)(1,5)(0,4)
      (-1,4)
uuuu\pscurve(0,4)(1,4)(1,3)(0.5,3)
      (0.5,2)(1,2)(1,1)(0,1)(0,0)
uuuu\psarc(0,2){2}{-90}{0}
uu}
uu\pscustom[fillstyle=crosshatch*,
      fillcolor=red]{\perimeter}
uu\psset{showpoints=true}
uu\rput(3,0){\perimeter}
\end{pspicture}
```

Sortie



Clipping

`\begin{psclip}{frontière}...\end{psclip}` pour dessiner uniquement ce qui est inclus dans la frontière.

Clipping

`\begin{psclip}{frontière}... \end{psclip}` pour dessiner uniquement ce qui est inclus dans la frontière.

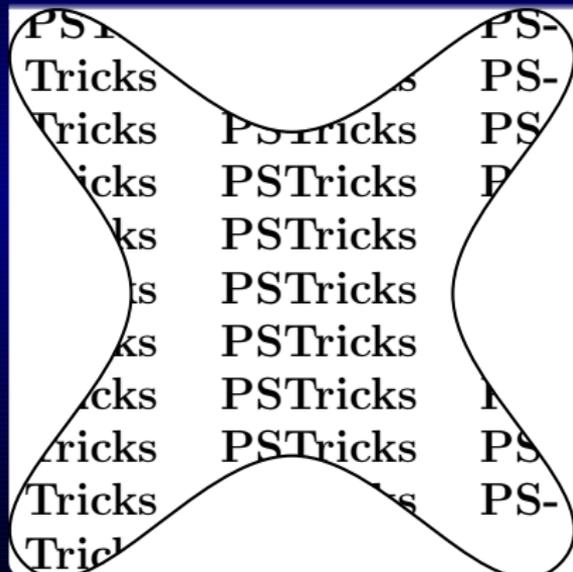
Code



Fichier Beamer3/code112.tex

```
\begin{pspicture}(0,-2.5)(5,2.5)
\begin{psclip}{
\psccurve(1,0)(0,-2.5)(2.5,-1.5)
(5,-2.5)(4,0)(5,2.5)(2.5,1.5)
(0,2.5)
}
\parbox{5cm}{%
\LARGE\bfseries PSTricks PSTricks
PSTricks PSTricks PSTricks
PSTricks
}
\end{psclip}
\end{pspicture}
```

Sortie



Effets de texte

- `\pstextpath[pos] (x,y){chemin}{texte}` trace le texte en suivant le chemin, aligné à la position `pos` (`l`, `c` ou `r`) par rapport à la longueur totale du chemin, et décalé de `(x,y)` par rapport à celui-ci;

Effets de texte

- `\pstextpath[pos](x,y){chemin}{texte}` trace le texte en suivant le chemin, aligné à la position `pos` (`l`, `c` ou `r`) par rapport à la longueur totale du chemin, et décalé de `(x,y)` par rapport à celui-ci;
- `\pscharpath[options]{texte}` affiche la région délimitée par texte.

Pavages

- `\multirput [référence]{rotation}(x0,y0)(dx,dy){n}{texte}` va placer n fois `texte` en translatant à chaque fois de (dx,dy) par rapport au point $(x0,y0)$;

Pavages

- `\multirput [référence]{rotation}(x0,y0)(dx,dy){n}{texte}` va placer n fois `texte` en translatant à chaque fois de (dx,dy) par rapport au point $(x0,y0)$;
- `\multips{rotation}(x0,y0)(dx,dy){n}{texte}` fera la même chose pour des objets de taille nulle.

Pavages

- `\multirput [référence]{rotation}(x0,y0)(dx,dy){n}{texte}` va placer n fois `texte` en translatant à chaque fois de (dx,dy) par rapport au point $(x0,y0)$;
- `\multips{rotation}(x0,y0)(dx,dy){n}{texte}` fera la même chose pour des objets de taille nulle.

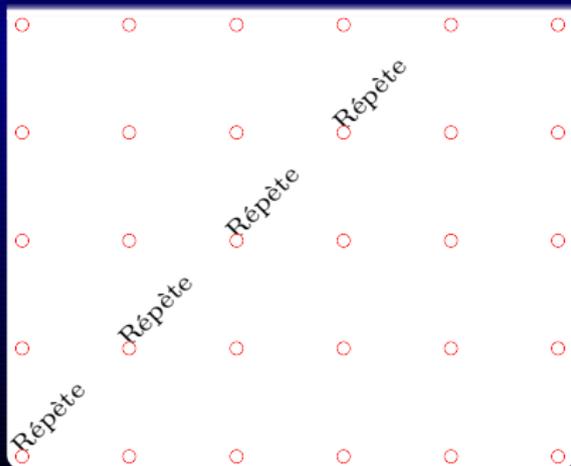
Code



Fichier Beamer3/code118.tex

```
\begin{pspicture}(0,0)(5,4)
uu\multirput[1B]{45}(1,1){4}{Répète}
uu\multirput(1,0){6}{
uuuu\multirput(0,1){5}{
uuuuuu\psdots[linecolor=red,dotstyle=
o](0,0)
uuuu}
uu}
\end{pspicture}
```

Sortie



Répétitions

`\multido{\variable=debut+delta}{n}{...}` va répéter n fois son dernier argument pour `\variable` variant de `debut` à `debut + (n - 1) · delta` (nécessite l'extension `multido`).

Répétitions

`\multido{\variable=debut+delta}{n}{...}` va répéter n fois son dernier argument pour `\variable` variant de `debut` à `debut + (n - 1) · delta` (nécessite l'extension `multido`).

Code

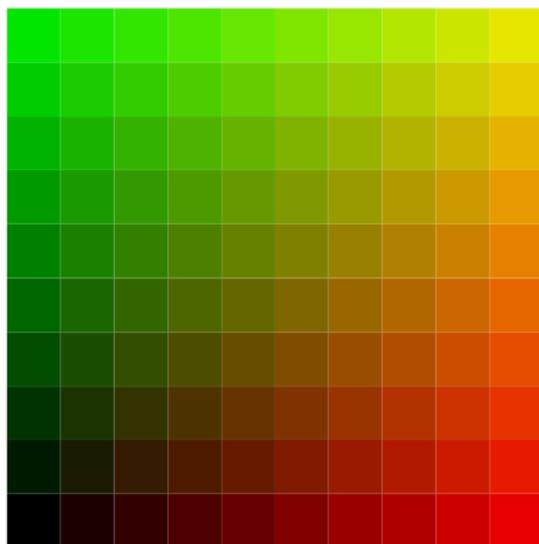


Fichier Beamer3/code120.tex

```
\psset{fillstyle=solid,linestyle=none}
}

\begin{pspicture}(0,0)(1,1)
  \multido{\nx=0.0+0.1}{10}{
    \multido{\ny=0.0+0.1}{10}{
      \newrgbcolor{c}{\nx}{\ny}
      \rput(\nx,\ny){
        \psframe[fillcolor=c](0,0)
          (0.1,0.1)
      }
    }
  }
\end{pspicture}
```

Sortie



Graphiques 3D avec pst-3dplot

... et encore beaucoup d'autres possibilités, consulter par exemple :

<http://tug.org/PSTricks/main.cgi?file=examples>

Graphiques 3D avec pst-3dplot

... et encore beaucoup d'autres possibilités, consulter par exemple :

<http://tug.org/PSTricks/main.cgi?file=examples>

Code

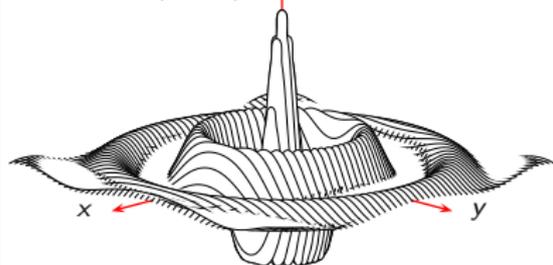


Fichier Beamer3/code122.tex

```
\begin{pspicture}(-2.5,-1)(2.5,2.5)
\psset{Beta=15,unit=0.45cm}
\rput[1](-5.5,4){$z=\frac{\cos 200\sqrt{x^2+y^2}}{0.3+0.3(x^2+y^2)}$}
\pstThreeDCoor[xMin=-1,xMax=5,yMin=-1,yMax=5,zMin=-1,zMax=5]
\psplotThreeD[yPlotpoints=50,
xPlotpoints=50,linewidth=0.1pt,
hiddenLine=true,drawStyle=xyLines,
plotstyle=curve,fillstyle=solid](-4,4)(-4,4){x2exp y2exp add
0.5exp 200 mul cos 0.3 0.3 x2 exp y2 exp add mul add div}
\end{pspicture}
```

Sortie

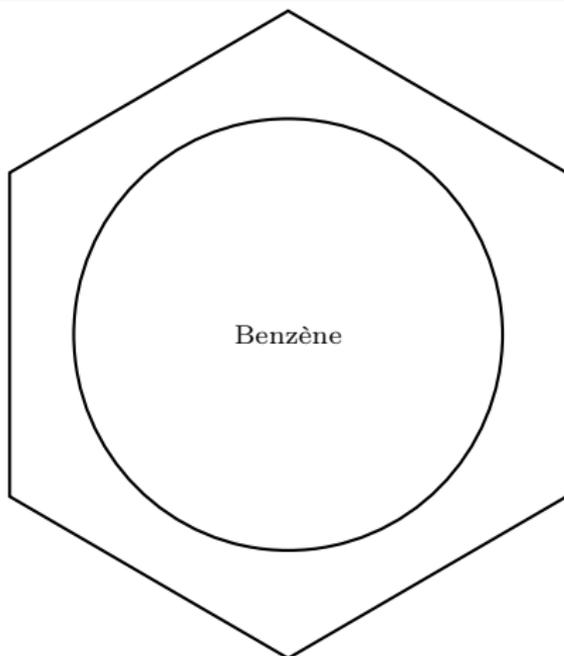
$$z = \frac{\cos 200\sqrt{x^2+y^2}}{0.3+0.3(x^2+y^2)}$$



Exercice

Trouver le code pour obtenir la sortie suivante

Sortie



Solution

Code



Fichier Beamer3/code123.tex

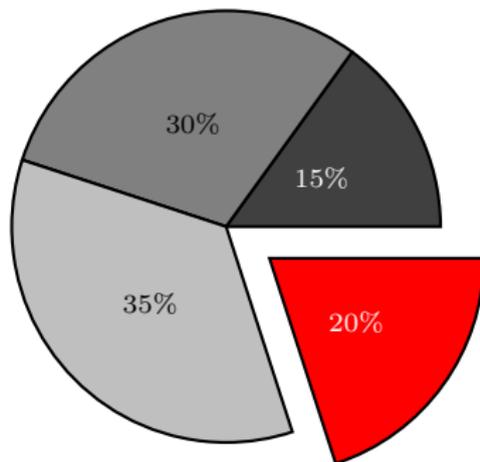
```
\SpecialCoor
\degrees [12]

\begin{pspicture}(-5,-3)(5,3)
  \pspolygon(3;1)(3;3)(3;5)(3;7)(3;9)(3;11)
  \pscircle{2}
  \rput(0,0){Benzène}
\end{pspicture}
```

Exercice

Trouver le code pour obtenir la sortie suivante

Sortie



Code



Fichier Beamer3/code124.tex

```
\SpecialCoor
\degrees [100]

\begin{pspicture}(-5,-3)(5,3)
  \psset{fillstyle=solid}

  \pswedge[fillcolor=darkgray](0,0){2}{0}{15}
  \rput(1;7.5){\white_15\%}

  \pswedge[fillcolor=gray](0,0){2}{15}{45}
  \rput(1;30){30\%}

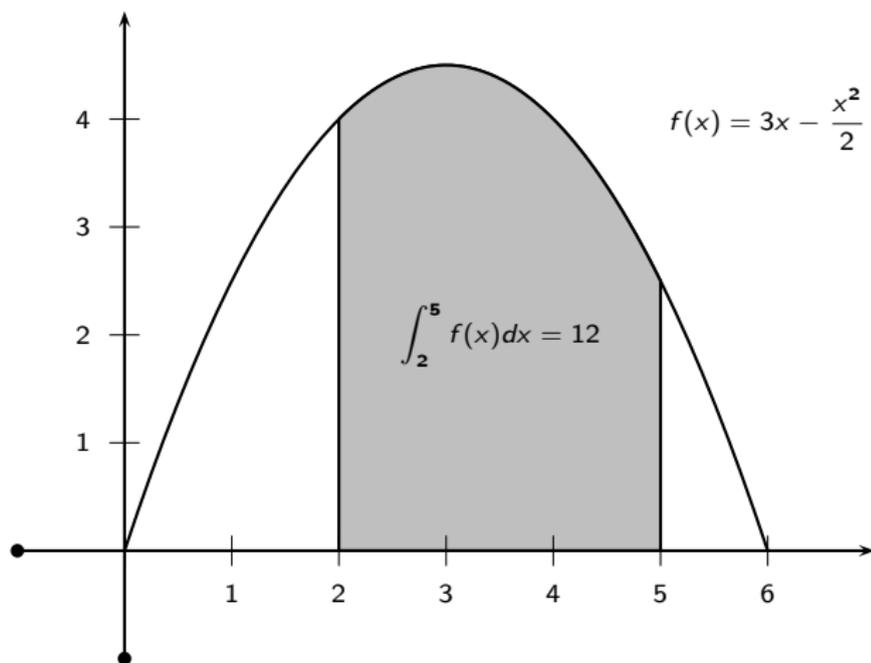
  \pswedge[fillcolor=lightgray](0,0){2}{45}{80}
  \rput(1;62.5){35\%}

  \rput(0.5;90){
    \pswedge[fillcolor=red](0,0){2}{80}{100}
    \rput(1;90){\white_20\%}
  }
\end{pspicture}
```

Exercice

Trouver le code pour obtenir la sortie suivante

Sortie



Code



Fichier Beamer3/code125.tex

```
\begin{pspicture}(-2,-1.2)(8,5.2)
uu\psplot[algebraic]{0}{6}{-0.5*x^2+3*x}
uu\pscustom[fillstyle=solid,fillcolor=lightgray]{
uuuu\psplot[algebraic]{2}{5}{-0.5*x^2+3*x}
uuuu\psline(5,0)(2,0)(2,4)
uu}
uu\psaxes{* ->}(0,0)(-1,-1)(7,5)
uu\rput(6,4){$\displaystyle f(x)=3x-\frac{x^2}{2}$}
uu\rput(3.5,2){$\displaystyle \int_{-2}^5 f(x) dx=12$}
\end{pspicture}
```

Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

Sortie



Code



Fichier Beamer3/code126.tex

```
\newrgbcolor{ORSAYBLUE}{0.15□0.32□0.57}

\begin{pspicture}(-10,-4)(10,4)
  \begin{psclip}{
    \pspolygon[linestyle=none](-3.5,0)(0,3.5)(3.5,0)(0,-3.5)
  }
  \multips(0,3.4)(0,-0.4){18}{
    \psframe*[linecolor=lightgray](-3.5,-0.1)(3.5,0.1)
  }
  \end{psclip}
  \psset{linecolor=ORSAYBLUE}
  \pswedge*(0,0.1){1.8}{0}{158}
  \pswedge*(0,0.1){1.8}{180}{338}
\end{pspicture}
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
 - Manipulation de fichiers PostScript et PDF
 - Listings et programmes informatiques
 - Partitions de musique
 - Lettres et courrier
 - Affiches et posters
 - Les extensions `pstricks`
 - **Transparents avec Seminar**
 - Présentations avec Beamer
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

La classe seminar

Permet de faire des transparents rapidement.

La classe seminar

Permet de faire des transparents rapidement.

La compilation se fait avec \LaTeX (et non Pdf \LaTeX) si l'on souhaite utiliser `semcolor`, qui repose sur `pstricks`.

La classe seminar

Permet de faire des transparents rapidement.

La compilation se fait avec \LaTeX (et non Pdf \LaTeX) si l'on souhaite utiliser semcolor, qui repose sur pstricks.

Chargement



Fichier Beamer3/code129.tex

```
\documentclass[a4]{seminar}

\usepackage{semcolor, slidesec, fancybox}
%Correction de bugs
\input{seminar.bug}
\input{seminar.bg2}

%pour gérer correctement les tailles et changements d'orientation de
page
\def\printlandscape{\special{papersize=297mm,210mm}}
```

La classe seminar

Nouveaux environnements :

- `\begin{slide}[largeur,hauteur]... \end{slide}` pour ajouter un transparent supplémentaire en mode « paysage » ;
- `\begin{slide*}[largeur,hauteur]... \end{slide*}` idem mais en mode « portrait » ;
- `\begin{note}... \end{note}` pour ajouter une note.

La classe seminar

Nouveaux environnements :

- `\begin{slide}[largeur,hauteur]... \end{slide}` pour ajouter un transparent supplémentaire en mode « paysage » ;
- `\begin{slide*}[largeur,hauteur]... \end{slide*}` idem mais en mode « portrait » ;
- `\begin{note}... \end{note}` pour ajouter une note.

Options de la classe :

- `notes` pour imprimer notes et transparents ;
- `slidesonly` pour n'imprimer que les transparents ;
- `notesonly` pour n'imprimer que les notes ;
- `article` pour imprimer un article dont le corps est formé du contenu des notes avec les transparents en figures flottantes.

Extensions à utiliser avec seminar

- L'extension `slidesec` :
 - `\slidechapter`, `\slideheading` et `\slidesubheading` pour définir des titres et sous-titres ;
 - `\listofslides` pour afficher une table des transparents ;
 - `\slidecontents` pour afficher le contenu des transparents ;
 - `\Slidecontents` pour afficher en plus l'endroit où l'on se trouve.

Extensions à utiliser avec seminar

- L'extension `slidesec` :
 - `\slidechapter`, `\slideheading` et `\slidesubheading` pour définir des titres et sous-titres ;
 - `\listofslides` pour afficher une table des transparents ;
 - `\slidecontents` pour afficher le contenu des transparents ;
 - `\Slidecontents` pour afficher en plus l'endroit où l'on se trouve.
- L'extension `semcolor` permet d'utiliser de la couleur.

Extensions à utiliser avec seminar

- L'extension `slidesec` :
 - `\slidechapter`, `\slideheading` et `\slidesubheading` pour définir des titres et sous-titres ;
 - `\listofslides` pour afficher une table des transparents ;
 - `\slidecontents` pour afficher le contenu des transparents ;
 - `\Slidecontents` pour afficher en plus l'endroit où l'on se trouve.
- L'extension `semcolor` permet d'utiliser de la couleur.
- L'extension `semhelv` permet d'utiliser une fonte sans sérif.

Extensions à utiliser avec seminar

- L'extension `slidesec` :
 - `\slidechapter`, `\slideheading` et `\slidesubheading` pour définir des titres et sous-titres ;
 - `\listofslides` pour afficher une table des transparents ;
 - `\slidecontents` pour afficher le contenu des transparents ;
 - `\Slidecontents` pour afficher en plus l'endroit où l'on se trouve.
- L'extension `semcolor` permet d'utiliser de la couleur.
- L'extension `semhelv` permet d'utiliser une fonte sans sérif.
- L'extension `semlayer` permet des effets spéciaux (superposition).

1

+

Liste des transparents

Types et classes de documents supplémentaires

- 2 Lettres et courriers
- 3 Transparents avec Seminar
- 4 Présentations avec Beamer

Programmation \LaTeX avancée

- 6 Tests et branchements conditionnels
- 8 Boucles
- 9 Commandes

+

+

`\listofslides`

7

+

Types et classes de documents supplémentaires

- ✓• Lettres et courriers 2
- ✓• Transparents avec Seminar 3
- ✓• Présentations avec Beamer 4

Programmation \LaTeX avancée

- ✓• Tests et branchements conditionnels 6
- ⇒• Boucles 8
- Commandes 9

+

+

`\Slidecontents`

Personnalisation

- `\slideframe{style}` pour la décoration du bord, avec `style = ⟨none, plain, shadow, double, oval⟩` (les 3 dernières nécessitent `fancybox`).
- `\slideframe*` superpose les styles.

Personnalisation

- `\slideframe{style}` pour la décoration du bord, avec `style = \langle none, plain, shadow, double, oval \rangle` (les 3 dernières nécessitent `fancybox`).
- `\slideframe*` superpose les styles.
- On peut colorier le fond avec les styles `scshadow`, `sdouble` ou `scoval` et en donnant des paramètres de remplissage avec `psset`.

Personnalisation

- `\slideframe{style}` pour la décoration du bord, avec `style = ⟨none, plain, shadow, double, oval⟩` (les 3 dernières nécessitent `fancybox`).
- `\slideframe*` superpose les styles.
- On peut colorier le fond avec les styles `scshadow`, `sdouble` ou `scoval` et en donnant des paramètres de remplissage avec `psset`.
- `\newslideframe{nom}{apparence}` pour définir un nouveau style, le contenu du transparent étant accessible dans `apparence` via `#1`.

Personnalisation

- `\slideframe{style}` pour la décoration du bord, avec `style = ⟨none, plain, shadow, double, oval⟩` (les 3 dernières nécessitent `fancybox`).
- `\slideframe*` superpose les styles.
- On peut colorier le fond avec les styles `scshadow`, `sdouble` ou `scoval` et en donnant des paramètres de remplissage avec `psset`.
- `\newslideframe{nom}{apparence}` pour définir un nouveau style, le contenu du transparent étant accessible dans `apparence` via `#1`.

Exemples



Fichier Beamer3/code133.tex

```
\slideframe[\psset{fillstyle=solid,fillcolor=red}]{scplain}

\slideframe[\psset{fillstyle=gradient,gradbegin=pink,gradend=yellow
}]{scshadow}

\newslideframe{CIES}{%
  \rput(1.5,-0.4){\psovalbox[fillstyle=solid,fillcolor=blue]{\white\
    LARGE\bfseries_CIES}}%
  #1%
}
```


Table des matières

- 1 Types de documents supplémentaires en \LaTeX
 - Manipulation de fichiers PostScript et PDF
 - Listings et programmes informatiques
 - Partitions de musique
 - Lettres et courrier
 - Affiches et posters
 - Les extensions pstricks
 - Transparents avec Seminar
 - **Présentations avec Beamer**
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Exemple



Fichier Beamer3/code134.tex

```
\documentclass[french]{beamer}

\begin{document}
  \begin{frame}
    \frametitle{Ma première page beamer}
    \framesubtitle{C'est très simple à écrire}
    Le contenu se centre verticalement dans la page, la fonte est
      sans serif et le texte est aligné à gauche. \bigskip

    \begin{itemize}
      \item Les listes
      \item à puces
      \item sont jolies
    \end{itemize} \bigskip

    \begin{enumerate}
      \item Les listes
      \item numérotées
      \item aussi
    \end{enumerate}
  \end{frame}
\end{document}
```

Ma première page beamer

C'est très simple à écrire

Le contenu se centre verticalement dans la page, la fonte est sans serif et le texte est aligné à gauche.

- Les listes
- à puces
- sont jolies

- 1 Les listes
- 2 numérotées
- 3 aussi

La page de titre

Code



Fichier Beamer3/code135.tex

```
\title{Mon premier document Beamer}  
\author{Jean Dupont}  
\institute{CIES}  
\date{\today}  
  
\begin{frame}  
  \titlepage  
  \begin{center}\url{http://jean-dupont.com}\end{center}  
\end{frame}
```

Mon premier document Beamer

Jean Dupont

CIES

10 avril 2009

<http://jean-dupont.com>

Colonnes

- `\begin{columns}[options]` permet de commencer un groupe de colonnes.

Colonnes

- `\begin{columns}[options]` permet de commencer un groupe de colonnes. Les options peuvent être :
 - `c` ou `T` pour aligner le milieu ou le haut des colonnes entre elles ;
 - `b` ou `t` pour aligner les lignes du bas ou du haut entre elles ;
 - `onlytextwidth` pour remplir toute la largeur et `totalwidth=largeur` pour remplir une largeur donnée (par défaut `totalwidth=\pagewidth`, c'est à dire plus que `\textwidth`).

Colonnes

- `\begin{columns}[options]` permet de commencer un groupe de colonnes. Les options peuvent être :
 - `c` ou `T` pour aligner le milieu ou le haut des colonnes entre elles ;
 - `b` ou `t` pour aligner les lignes du bas ou du haut entre elles ;
 - `onlytextwidth` pour remplir toute la largeur et `totalwidth=largeur` pour remplir une largeur donnée (par défaut `totalwidth=\pagewidth`, c'est à dire plus que `\textwidth`).
- `\begin{column}{largeur}` à l'intérieur d'un environnement `columns` crée une nouvelle colonne.

Colonnes

- `\begin{columns}[options]` permet de commencer un groupe de colonnes. Les options peuvent être :
 - `c` ou `T` pour aligner le milieu ou le haut des colonnes entre elles ;
 - `b` ou `t` pour aligner les lignes du bas ou du haut entre elles ;
 - `onlytextwidth` pour remplir toute la largeur et `totalwidth=largeur` pour remplir une largeur donnée (par défaut `totalwidth=\pagewidth`, c'est à dire plus que `\textwidth`).
- `\begin{column}{largeur}` à l'intérieur d'un environnement `columns` crée une nouvelle colonne.

Code



Fichier Beamer3/code140.tex

```
\begin{columns}[t, onlytextwidth]
  \begin{column}{0.2\textwidth}La
    première colonne\end{column}
  \begin{column}{0.3\textwidth}La
    deuxième colonne\end{column}
  \begin{column}{0.4\textwidth}La
    troisième colonne\end{column}
\end{columns}
```

Sortie

La pre- mière colonne	La deuxième colonne	La troisième colonne
-----------------------------	---------------------------	-------------------------

Code



Fichier Beamer3/code141.tex

```
\begin{block}{Un titre normal}
  Pour un bloc normal
\end{block}
\begin{block}{Un bloc de blocs}
  On peut imbriquer sans problème blocs et colonnes, les théorèmes
  et preuves ont aussi leurs propres blocs.
  \begin{columns}
    \begin{column}{0.45\textwidth}
      \begin{theorem} Ceci n'est pas une négation en bloc. \end{theorem}
    \begin{proof} C'est évident! \end{proof}
    \end{column}
    \begin{column}{0.45\textwidth}
      \begin{exampleblock}{Un bloc d'exemple} Pour les illustrations \end{exampleblock}
    \begin{alertblock}{Un bloc d'alerte!} Pour les choses importantes \end{alertblock}
    \end{column}
  \end{columns}
\end{block}
```

Un titre normal

Pour un bloc normal

Un bloc de blocs

On peut imbriquer sans problème blocs et colonnes, les théorèmes et preuves ont aussi leurs propres blocs.

Théorème

Ceci n'est pas une négation en bloc.

Un bloc d'exemple

Pour les illustrations

Démonstration.

C'est évident !

Un bloc d'alerte !

Pour les choses importantes

Thèmes de Beamer

On active un thème général avec `\usetheme[options]{theme}` dans le préambule.

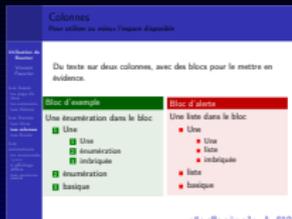
Thèmes de Beamer

On active un thème général avec `\usetheme[options]{theme}` dans le préambule.

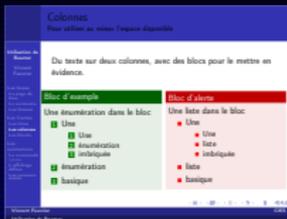
Ce sera une combinaison de quatre sous-thèmes qui peuvent aussi être activés individuellement :

- `\usecolortheme[options]{theme}` pour le thème de couleur (qui peut être global, interne ou externe) ;
- `\useinnertheme[options]{theme}` pour le thème des éléments internes (blocs, énumérations, etc. . .) ;
- `\useoutertheme[options]{theme}` pour le thème des éléments externes (barre de navigation, foliation, etc. . .) ;
- `\usefonttheme[options]{theme}` pour le thème de fontes.

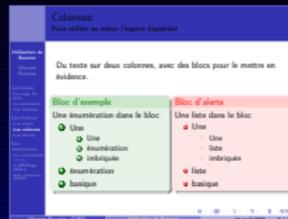
Thèmes généraux, avec sidebar en thème externe



Antibes



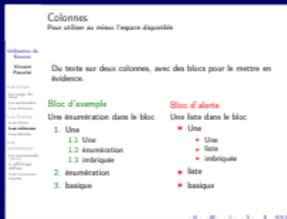
Berlin



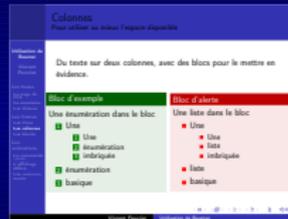
Boadilla



CambridgeUS



default



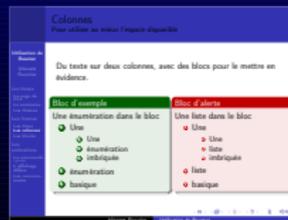
Luebeck



Madrid



Pittsburgh



Warsaw

Thèmes de couleur complets, avec sidebar en thème externe

Colonne
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

1. Une
- 1.1 énumération
- 1.2 indubitable

Bloc d'alerte
Une liste dans le bloc

- Une
- Une
- liste
- indubitable
- liste
- basique

albatross

Colonne
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

1. Une
- 1.1 Une
- 1.2 énumération
- 1.3 indubitable

Bloc d'alerte
Une liste dans le bloc

- Une
- Une
- liste
- indubitable
- liste
- basique

beaver

Colonne
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

1. Une
- 1.1 Une
- 1.2 énumération
- 1.3 indubitable

Bloc d'alerte
Une liste dans le bloc

- Une
- Une
- liste
- indubitable
- liste
- basique

beetle

Colonne
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

1. Une
- 1.1 Une
- 1.2 énumération
- 1.3 indubitable

Bloc d'alerte
Une liste dans le bloc

- Une
- Une
- liste
- indubitable
- liste
- basique

crane

Colonne
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

1. Une
- 1.1 Une
- 1.2 énumération
- 1.3 indubitable

Bloc d'alerte
Une liste dans le bloc

- Une
- Une
- liste
- indubitable
- liste
- basique

default

Colonne
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

1. Une
- 1.1 Une
- 1.2 énumération
- 1.3 indubitable

Bloc d'alerte
Une liste dans le bloc

- Une
- Une
- liste
- indubitable
- liste
- basique

dove

Colonne
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

- 1.1 Une
- 1.2 énumération
- 1.3 indubitable

Bloc d'alerte
Une liste dans le bloc

- Une
- Une
- liste
- indubitable
- liste
- basique

fly

Colonne
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

1. Une
- 1.1 Une
- 1.2 énumération
- 1.3 indubitable

Bloc d'alerte
Une liste dans le bloc

- Une
- Une
- liste
- indubitable
- liste
- basique

seagull

Colonne
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

1. Une
- 1.1 Une
- 1.2 énumération
- 1.3 indubitable

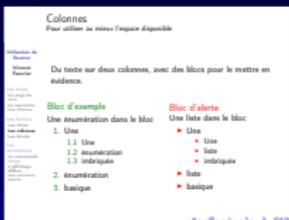
Bloc d'alerte
Une liste dans le bloc

- Une
- Une
- liste
- indubitable
- liste
- basique

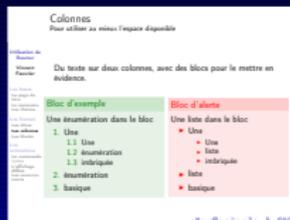
wolverine

Thèmes de couleur partiels, avec sidebar en thème externe

• Thèmes de couleur internes



lily

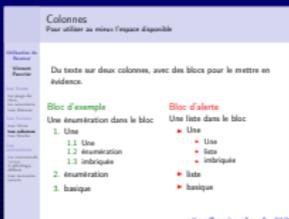


orchid

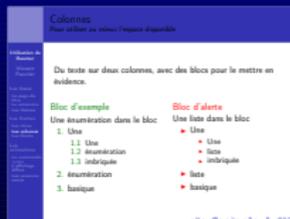


rose

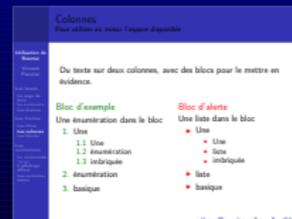
• Thèmes de couleur externes



whale



seahorse



dolphin

Thème de couleur « structure » avec une option pour la couleur de base

Colonne

Pour utiliser au mieux l'espace disponible

Utilisation de Beamer

Vicent Fauvier

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple

Une énumération dans le bloc

1. Une
 - 1.1 Une
 - 1.2 énumération
 - 1.3 imbriquée
2. énumération
3. basique

Bloc d'alerte

Une liste dans le bloc

- ▶ Une
 - ▶ Une
 - ▶ liste
 - ▶ imbriquée
- ▶ liste
- ▶ basique

Les titres
Le page de titre
La navigation
Les diapositives
Les titres
Les colonnes
Les blocs
Les commandes
La navigation
Le langage
Le langage
Les commandes
Les titres

[named=violet]

Colonne

Pour utiliser au mieux l'espace disponible

Utilisation de Beamer

Vicent Fauvier

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple

Une énumération dans le bloc

1. Une
 - 1.1 Une
 - 1.2 énumération
 - 1.3 imbriquée
2. énumération
3. basique

Bloc d'alerte

Une liste dans le bloc

- ▶ Une
 - ▶ Une
 - ▶ liste
 - ▶ imbriquée
- ▶ liste
- ▶ basique

Les titres
Le page de titre
La navigation
Les diapositives
Les titres
Les colonnes
Les blocs
Les commandes
La navigation
Le langage
Le langage
Les commandes
Les titres

[named=green]

Colonne

Pour utiliser au mieux l'espace disponible

Utilisation de Beamer

Vicent Fauvier

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple

Une énumération dans le bloc

1. Une
 - 1.1 Une
 - 1.2 énumération
 - 1.3 imbriquée
2. énumération
3. basique

Bloc d'alerte

Une liste dans le bloc

- ▶ Une
 - ▶ Une
 - ▶ liste
 - ▶ imbriquée
- ▶ liste
- ▶ basique

Les titres
Le page de titre
La navigation
Les diapositives
Les titres
Les colonnes
Les blocs
Les commandes
La navigation
Le langage
Le langage
Les commandes
Les titres

[named=purple]

Colonne

Pour utiliser au mieux l'espace disponible

Utilisation de Beamer

Vicent Fauvier

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple

Une énumération dans le bloc

1. Une
 - 1.1 Une
 - 1.2 énumération
 - 1.3 imbriquée
2. énumération
3. basique

Bloc d'alerte

Une liste dans le bloc

- ▶ Une
 - ▶ Une
 - ▶ liste
 - ▶ imbriquée
- ▶ liste
- ▶ basique

Les titres
Le page de titre
La navigation
Les diapositives
Les titres
Les colonnes
Les blocs
Les commandes
La navigation
Le langage
Le langage
Les commandes
Les titres

[named=brown]

Thèmes internes, avec orchid pour les couleurs

Colonnes
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple	Bloc d'alerte
Une énumération dans le bloc	Une liste dans le bloc
<ul style="list-style-type: none">1. Une2. énumération3. imbriquée	<ul style="list-style-type: none">• Une• Une• liste• imbriquée
<ul style="list-style-type: none">• énumération• basique	<ul style="list-style-type: none">• liste• basique

circles

Colonnes
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple	Bloc d'alerte
Une énumération dans le bloc	Une liste dans le bloc
<ol style="list-style-type: none">1. Une1.1. énumération1.2. imbriquée	<ul style="list-style-type: none">• Une• Une• liste• imbriquée
<ol style="list-style-type: none">2. énumération3. basique	<ul style="list-style-type: none">• liste• basique

default

Colonnes
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple	Bloc d'alerte
Une énumération dans le bloc	Une liste dans le bloc
<ul style="list-style-type: none">• Une• énumération• imbriquée	<ul style="list-style-type: none">• Une• Une• liste• imbriquée
<ul style="list-style-type: none">• énumération• basique	<ul style="list-style-type: none">• liste• basique

inmargin

Colonnes
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple	Bloc d'alerte
Une énumération dans le bloc	Une liste dans le bloc
<ul style="list-style-type: none">1. Une2. énumération3. imbriquée	<ul style="list-style-type: none">• Une• Une• liste• imbriquée
<ul style="list-style-type: none">• énumération• basique	<ul style="list-style-type: none">• liste• basique

rectangles

Colonnes
Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

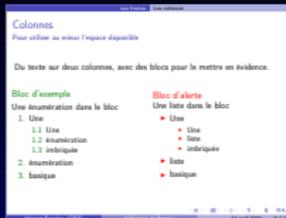
Bloc d'exemple	Bloc d'alerte
Une énumération dans le bloc	Une liste dans le bloc
<ul style="list-style-type: none">• Une• énumération• imbriquée	<ul style="list-style-type: none">• Une• Une• liste• imbriquée
<ul style="list-style-type: none">• énumération• basique	<ul style="list-style-type: none">• liste• basique

rounded

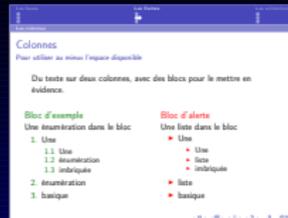
Thèmes externes, avec dolphin pour les couleurs



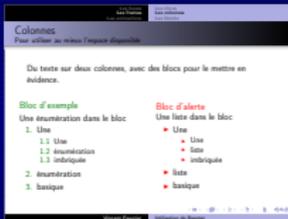
default



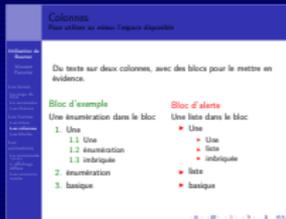
infolines



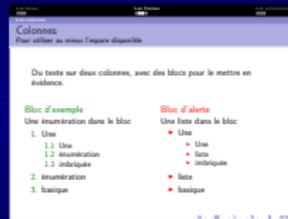
miniframes



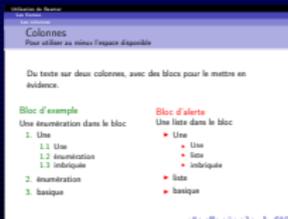
shadow



sidebar



smoothbars



smoothtree



split



tree

Thèmes de fontes

Colonnes

Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

- 1. Une
 - 1.1 Une
 - 1.2 énumération
 - 1.3 imbriquée
- 2. énumération
- 3. basique

Bloc d'alerte
Une liste dans le bloc

- ▶ Une
 - ▶ Une
 - ▶ liste
 - ▶ imbriquée
- ▶ liste
- ▶ basique



default

Colonnes

Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

- 1. Une
 - 1.1 Une
 - 1.2 énumération
 - 1.3 imbriquée
- 2. énumération
- 3. basique

Bloc d'alerte
Une liste dans le bloc

- ▶ Une
 - ▶ Une
 - ▶ liste
 - ▶ imbriquée
- ▶ liste
- ▶ basique



professional

Colonnes

Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

- 1. Une
 - 1.1 Une
 - 1.2 énumération
 - 1.3 imbriquée
- 2. énumération
- 3. basique

Bloc d'alerte
Une liste dans le bloc

- ▶ Une
 - ▶ Une
 - ▶ liste
 - ▶ imbriquée
- ▶ liste
- ▶ basique



serif

Colonnes

Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

- 1. Une
 - 1.1 Une
 - 1.2 énumération
 - 1.3 imbriquée
- 2. énumération
- 3. basique

Bloc d'alerte
Une liste dans le bloc

- ▶ Une
 - ▶ Une
 - ▶ liste
 - ▶ imbriquée
- ▶ liste
- ▶ basique



structurebold

Colonnes

Pour utiliser au mieux l'espace disponible

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

Bloc d'exemple
Une énumération dans le bloc

- 1. Une
 - 1.1 Une
 - 1.2 énumération
 - 1.3 imbriquée
- 2. énumération
- 3. basique

Bloc d'alerte
Une liste dans le bloc

- ▶ Une
 - ▶ Une
 - ▶ liste
 - ▶ imbriquée
- ▶ liste
- ▶ basique



structureitalicserif

COLONNES

POUR UTILISER AU MEILLEUR L'ESPACE DISPONIBLE

Du texte sur deux colonnes, avec des blocs pour le mettre en évidence.

BLOC D'EXEMPLE
Une énumération dans le bloc

- 1. Une
 - 1.1 Une
 - 1.2 énumération
 - 1.3 imbriquée
- 2. énumération
- 3. basique

BLOC D'ALERTE
Une liste dans le bloc

- ▶ Une
 - ▶ Une
 - ▶ liste
 - ▶ imbriquée
- ▶ liste
- ▶ basique



structuresmallcapserif

Personnalisation du thème

- `\setbeamercolor{élément}{attribut=couleur}` permet de changer la couleur d'un attribut (fg, bg...) d'un élément (background canvas, normal text, alerted text...):

Personnalisation du thème

- `\setbeamercolor{élément}{attribut=couleur}` permet de changer la couleur d'un attribut (fg, bg...) d'un élément (background canvas, normal text, alerted text...):
 - on peut donner une couleur par son nom, préexistant ou défini par `\definecolor{couleur}{rgb}{R,G,B}`,
 - par une combinaison de deux couleurs : `red!20!black` pour un mélange 20% rouge/80% noir;

Personnalisation du thème

- `\setbeamercolor{élément}{attribut=couleur}` permet de changer la couleur d'un attribut (fg, bg...) d'un élément (background canvas, normal text, alerted text...):
 - on peut donner une couleur par son nom, préexistant ou défini par `\definecolor{couleur}{rgb}{R,G,B}`,
 - par une combinaison de deux couleurs : `red!20!black` pour un mélange 20% rouge/80% noir;
- `\setbeamertemplate{élément}modèle` permet d'utiliser un modèle prédéfini (`[vertical shading][options]`, `[square]`, `[circle]{rayon}...`);

Personnalisation du thème

- `\setbeamercolor{élément}{attribut=couleur}` permet de changer la couleur d'un attribut (fg, bg...) d'un élément (background canvas, normal text, alerted text...):
 - on peut donner une couleur par son nom, préexistant ou défini par `\definecolor{couleur}{rgb}{R,G,B}`,
 - par une combinaison de deux couleurs : `red!20!black` pour un mélange 20% rouge/80% noir;
- `\setbeamertemplate{élément}modèle` permet d'utiliser un modèle prédéfini (`[vertical shading][options]`, `[square]`, `[circle]{rayon}...`);
- `\setbeamerfont{élément}{attribut=valeur}` permet de changer un attribut (family, series, size) de fonte.

Personnalisation du thème

- `\setbeamercolor{élément}{attribut=couleur}` permet de changer la couleur d'un attribut (fg, bg...) d'un élément (background canvas, normal text, alerted text...):
 - on peut donner une couleur par son nom, préexistant ou défini par `\definecolor{couleur}{rgb}{R,G,B}`,
 - par une combinaison de deux couleurs : `red!20!black` pour un mélange 20% rouge/80% noir;
- `\setbeamertemplate{élément}modèle` permet d'utiliser un modèle prédéfini (`[vertical shading][options]`, `[square]`, `[circle]{rayon}...`);
- `\setbeamerfont{élément}{attribut=valeur}` permet de changer un attribut (family, series, size) de fonte.

Exemple



Fichier Beamer3/code150.tex

```
%Retour aux valeurs par défaut
\setbeamertemplate{background canvas}[default]
%Changement de la couleur de fond
\setbeamercolor{normal text}{bg=red!30!black}
```

Personnalisation du thème

- `\setbeamercolor{élément}{attribut=couleur}` permet de changer la couleur d'un attribut (fg, bg...) d'un élément (background canvas, normal text, alerted text...):
 - on peut donner une couleur par son nom, préexistant ou défini par `\definecolor{couleur}{rgb}{R,G,B}`,
 - par une combinaison de deux couleurs : `red!20!black` pour un mélange 20% rouge/80% noir;
- `\setbeamertemplate{élément}modèle` permet d'utiliser un modèle prédéfini (`[vertical shading][options]`, `[square]`, `[circle]{rayon}...`);
- `\setbeamerfont{élément}{attribut=valeur}` permet de changer un attribut (family, series, size) de fonte.

Exemple



Fichier Beamer3/code153.tex

```
%Retour aux valeurs par défaut
\setbeamertemplate{background canvas}[default]
%Changement du modèle de fond
\setbeamertemplate{background canvas}[vertical shading][bottom=
orange!40!black,middle=pink!60!black,top=blue!50!black]
```


Affichage différé

- `\pause` affiche uniquement ce qui précède sur la page en cours, puis l'affiche de nouveau suivi du reste sur les pages suivantes ;
- `\item<pages>` affiche un élément de liste uniquement sur les pages spécifiées ;
- `\only<pages>{texte}` affiche le texte uniquement sur les pages spécifiées.

Code



Fichier Beamer3/code155.tex

```
Voici\only<1>{\ldots}
\pause
une liste\only<-5>{incomplète}\only
<6>{complète}:
\pause
\begin{itemize}
  \item Premier élément
  \item<4-> Deuxième élément
  \item<5,6> Troisième élément
  \item<6> Quatrième élément
  \item Dernier élément (toujours
    visible)
\end{itemize}
\pause
Et voilà du texte.
```

Sortie

Voici une liste incomplète :

Affichage différé

- `\pause` affiche uniquement ce qui précède sur la page en cours, puis l'affiche de nouveau suivi du reste sur les pages suivantes ;
- `\item<pages>` affiche un élément de liste uniquement sur les pages spécifiées ;
- `\only<pages>{texte}` affiche le texte uniquement sur les pages spécifiées.

Code



Fichier Beamer3/code156.tex

```
Voici\only<1>{\ldots}
\pause
une liste\only<-5>{incomplète}\only
<6>{complète}:
\pause
\begin{itemize}
  \item Premier élément
  \item<4-> Deuxième élément
  \item<5,6> Troisième élément
  \item<6> Quatrième élément
  \item Dernier élément (toujours
    visible)
\end{itemize}
\pause
Et voilà du texte.
```

Sortie

Voici une liste incomplète :

- Premier élément

- Dernier élément (toujours visible)

Affichage différé

- `\pause` affiche uniquement ce qui précède sur la page en cours, puis l'affiche de nouveau suivi du reste sur les pages suivantes ;
- `\item<pages>` affiche un élément de liste uniquement sur les pages spécifiées ;
- `\only<pages>{texte}` affiche le texte uniquement sur les pages spécifiées.

Code



Fichier Beamer3/code157.tex

```
Voici\only<1>{\ldots}
\pause
une\liste\only<-5>{incomplète}\only
  <6>{complète}:
\pause
\begin{itemize}
\item\itemize
\item<4->\Deuxième\élément
\item<5,6>\Troisième\élément
\item<6>\Quatrième\élément
\item\ Dernier\élément (toujours\
  visible)
\end{itemize}
\pause
Et\voilà\du\texte.
```

Sortie

Voici une liste incomplète :

- Premier élément
- Deuxième élément

- Dernier élément (toujours visible)

Et voilà du texte.

Affichage différé

Conventions de désignation des pages d'un frame

- si on ne met rien (par exemple `\only{texte}`) alors implicitement ce sera toutes les pages ;

Affichage différé

Conventions de désignation des pages d'un frame

- si on ne met rien (par exemple `\only{texte}`) alors implicitement ce sera toutes les pages ;
- un nombre tout seul désigne un numéro de page ;

Affichage différé

Conventions de désignation des pages d'un frame

- si on ne met rien (par exemple `\only{texte}`) alors implicitement ce sera toutes les pages ;
- un nombre tout seul désigne un numéro de page ;
- + désigne la page en cours la première fois, et la page qui suit à chaque nouvelle utilisation ;

Affichage différé

Conventions de désignation des pages d'un frame

- si on ne met rien (par exemple `\only{texte}`) alors implicitement ce sera toutes les pages ;
- un nombre tout seul désigne un numéro de page ;
- + désigne la page en cours la première fois, et la page qui suit à chaque nouvelle utilisation ;
- $x-y$ désigne toutes les pages dont les numéros sont compris entre x et y ;

Affichage différé

Conventions de désignation des pages d'un frame

- si on ne met rien (par exemple `\only{texte}`) alors implicitement ce sera toutes les pages ;
- un nombre tout seul désigne un numéro de page ;
- + désigne la page en cours la première fois, et la page qui suit à chaque nouvelle utilisation ;
- x - y désigne toutes les pages dont les numéros sont compris entre x et y ;
- $-x$ désigne toutes les pages de 1 à x ;
- x - désigne toutes les pages à partir de x jusqu'à la fin du frame ;

Affichage différé

Conventions de désignation des pages d'un frame

- si on ne met rien (par exemple `\only{texte}`) alors implicitement ce sera toutes les pages ;
- un nombre tout seul désigne un numéro de page ;
- + désigne la page en cours la première fois, et la page qui suit à chaque nouvelle utilisation ;
- x - y désigne toutes les pages dont les numéros sont compris entre x et y ;
- $-x$ désigne toutes les pages de 1 à x ;
- x - désigne toutes les pages à partir de x jusqu'à la fin du frame ;
- on sépare les pages ou groupes de pages par des virgules ;
- le nombre total de pages du frame sera augmenté pour être au moins aussi grand que la dernière page désignée.

Affichage différé

Conventions de désignation des pages d'un frame

- si on ne met rien (par exemple `\only{texte}`) alors implicitement ce sera toutes les pages ;
- un nombre tout seul désigne un numéro de page ;
- + désigne la page en cours la première fois, et la page qui suit à chaque nouvelle utilisation ;
- x-y désigne toutes les pages dont les numéros sont compris entre x et y ;
- -x désigne toutes les pages de 1 à x ;
- x- désigne toutes les pages à partir de x jusqu'à la fin du frame ;
- on sépare les pages ou groupes de pages par des virgules ;
- le nombre total de pages du frame sera augmenté pour être au moins aussi grand que la dernière page désignée.

Code



Fichier Beamer3/code166.tex

```
\begin{itemize}
  \item<+>Premier élément
  \item<+>Deuxième élément
  \item<+>Troisième élément
  \item<+>Quatrième élément
\end{itemize}
```

Sortie

- Premier élément

Affichage différé

Conventions de désignation des pages d'un frame

- si on ne met rien (par exemple `\only{texte}`) alors implicitement ce sera toutes les pages ;
- un nombre tout seul désigne un numéro de page ;
- + désigne la page en cours la première fois, et la page qui suit à chaque nouvelle utilisation ;
- x-y désigne toutes les pages dont les numéros sont compris entre x et y ;
- -x désigne toutes les pages de 1 à x ;
- x- désigne toutes les pages à partir de x jusqu'à la fin du frame ;
- on sépare les pages ou groupes de pages par des virgules ;
- le nombre total de pages du frame sera augmenté pour être au moins aussi grand que la dernière page désignée.

Code



Fichier Beamer3/code167.tex

```
\begin{itemize}
  \item<+>Premier élément
  \item<+>Deuxième élément
  \item<+>Troisième élément
  \item<+>Quatrième élément
\end{itemize}
```

Sortie

- Premier élément
- Deuxième élément

Affichage différé

Conventions de désignation des pages d'un frame

- si on ne met rien (par exemple `\only{texte}`) alors implicitement ce sera toutes les pages ;
- un nombre tout seul désigne un numéro de page ;
- + désigne la page en cours la première fois, et la page qui suit à chaque nouvelle utilisation ;
- x-y désigne toutes les pages dont les numéros sont compris entre x et y ;
- -x désigne toutes les pages de 1 à x ;
- x- désigne toutes les pages à partir de x jusqu'à la fin du frame ;
- on sépare les pages ou groupes de pages par des virgules ;
- le nombre total de pages du frame sera augmenté pour être au moins aussi grand que la dernière page désignée.

Code



Fichier Beamer3/code168.tex

```
\begin{itemize}
  \item<+>Premier élément
  \item<+>Deuxième élément
  \item<+>Troisième élément
  \item<+>Quatrième élément
\end{itemize}
```

Sortie

- Premier élément
- Troisième élément

Affichage différé

Conventions de désignation des pages d'un frame

- si on ne met rien (par exemple `\only{texte}`) alors implicitement ce sera toutes les pages ;
- un nombre tout seul désigne un numéro de page ;
- + désigne la page en cours la première fois, et la page qui suit à chaque nouvelle utilisation ;
- x-y désigne toutes les pages dont les numéros sont compris entre x et y ;
- -x désigne toutes les pages de 1 à x ;
- x- désigne toutes les pages à partir de x jusqu'à la fin du frame ;
- on sépare les pages ou groupes de pages par des virgules ;
- le nombre total de pages du frame sera augmenté pour être au moins aussi grand que la dernière page désignée.

Code



Fichier Beamer3/code169.tex

```
\begin{itemize}
  \item<+>Premier élément
  \item<+>Deuxième élément
  \item<+>Troisième élément
  \item<+>Quatrième élément
\end{itemize}
```

Sortie

- Premier élément
- Troisième élément
- Quatrième élément

Affichage différé

Commandes et environnements acceptant des paramètres d'affichage différé

- `\emph`, `\textit`, `\textbf`, ...
- `\alert` pour souligner du texte, `\structure` pour le mode menu ;
- `\color`, `\includegraphics` ;
- `\label` pour étiqueter une page particulière d'un frame ;
- les colonnes et les environnements de blocs (`block`, `alertblock`, ...).

Code



Fichier Beamer3/code170.tex

```
Ce_\alert<2>{texte}_\color<2>{blue}
  est_\textbf<2>{torturé}_à_la_\
  textit<2>{page}_\pageref{torture
  }.\label<2>{torture}

\begin{columns}
  \begin{column}{0.45\textwidth}
    \begin{alertblock}<2>{Bloc
      différé}\end{alertblock}
  \end{column}
  \begin{column}<2>{0.45\textwidth}
    \begin{exampleblock}{Colonne
      différée}\end{exampleblock}
  \end{column}
\end{columns}
```

Sortie

Ce texte est torturé à la page 229.

Affichage différé

Commandes et environnements acceptant des paramètres d'affichage différé

- `\emph`, `\textit`, `\textbf`, ...
- `\alert` pour souligner du texte, `\structure` pour le mode menu ;
- `\color`, `\includegraphics` ;
- `\label` pour étiqueter une page particulière d'un frame ;
- les colonnes et les environnements de blocs (`block`, `alertblock`, ...).

Code



Fichier Beamer3/code172.tex

```
Ce_\alert<2>{texte}_\color<2>{blue}
  est_\textbf<2>{torturé}_à_la_\
  textit<2>{page}_\pageref{torture
  }.\label<2>{torture}
\begin{columns}
  \begin{column}{0.45\textwidth}
    \begin{alertblock}<2>{Bloc
      différé}\end{alertblock}
  \end{column}
  \begin{column}<2>{0.45\textwidth}
    \begin{exampleblock}{Colonne
      différée}\end{exampleblock}
  \end{column}
\end{columns}
```

Sortie

Ce **texte** est **torturé** à la *page 229*.

Bloc différé

**Colonne
différée**

Affichage différé

Incrémentation automatique

Les environnements `frame`, `itemize` et `enumerate` peuvent prendre le paramètre optionnel [`<+>`] :

Code



Fichier Beamer3/code174.tex

```
\begin{itemize}[<+>]  
  \item Voici  
  \item un affichage  
  \item progressif  
\end{itemize}  
\begin{enumerate}[<+>]  
  \item facile  
  \item à réaliser.  
\end{enumerate}
```

Sortie

- Voici

Affichage différé

Incrémentation automatique

Les environnements `frame`, `itemize` et `enumerate` peuvent prendre le paramètre optionnel `[<+>]` :

Code



Fichier Beamer3/code175.tex

```
\begin{itemize}[<+>]  
  \item Voici  
  \item un affichage  
  \item progressif  
\end{itemize}  
\begin{enumerate}[<+>]  
  \item facile  
  \item à réaliser.  
\end{enumerate}
```

Sortie

- Voici
- un affichage

Affichage différé

Incrémentation automatique

Les environnements `frame`, `itemize` et `enumerate` peuvent prendre le paramètre optionnel `[<+>]` :

Code



Fichier Beamer3/code176.tex

```
\begin{itemize}[<+>]
  \item Voici
  \item un affichage
  \item progressif
\end{itemize}
\begin{enumerate}[<+>]
  \item facile
  \item à réaliser.
\end{enumerate}
```

Sortie

- Voici
- un affichage
- progressif

Affichage différé

Incrémentation automatique

Les environnements `frame`, `itemize` et `enumerate` peuvent prendre le paramètre optionnel `[<+>]` :

Code



Fichier Beamer3/code177.tex

```
\begin{itemize}[<+>]
  \item Voici
  \item un affichage
  \item progressif
\end{itemize}
\begin{enumerate}[<+>]
  \item facile
  \item à réaliser.
\end{enumerate}
```

Sortie

- Voici
- un affichage
- progressif
- facile

Affichage différé

Incrémentation automatique

Les environnements `frame`, `itemize` et `enumerate` peuvent prendre le paramètre optionnel `[<+>]` :

Code



Fichier Beamer3/code178.tex

```
\begin{itemize}[<+>]
  \item Voici
  \item un affichage
  \item progressif
\end{itemize}
\begin{enumerate}[<+>]
  \item facile
  \item à réaliser.
\end{enumerate}
```

Sortie

- Voici
- un affichage
- progressif
- 1 facile
- 2 à réaliser.

Animation et superposition

- `\uncover<pages>{...}` s'utilise comme `\only`, mais conserve l'espace occupé par les éléments cachés;
- `\onslide<pages>` fait la même chose mais s'applique à tout ce qui suit (pas besoin d'accolades).

Code



Fichier Beamer3/code179.tex

```
Ce_texte_\only<2>{n'}est_\only<2>{pas_\ncache}.
```

```
Ce_texte_\uncover<2>{n'}est_\onslide<2>pas_\onslide_\ncache.
```

Sortie

Ce texte est caché.

Ce texte est caché.

Animation et superposition

- `\uncover<pages>{...}` s'utilise comme `\only`, mais conserve l'espace occupé par les éléments cachés;
- `\onslide<pages>` fait la même chose mais s'applique à tout ce qui suit (pas besoin d'accolades).

Code



Fichier Beamer3/code180.tex

```
Ce_texte_\only<2>{n'}est_\only<2>{pas_\ncache}.
```

```
Ce_texte_\uncover<2>{n'}est_\onslide<2>pas_\onslide_\ncache.
```

Sortie

Ce texte n'est pas caché.

Ce texte n'est pas caché.

Animation et superposition

Effets de transparence

`\setbeamercovered{mode}` permet de changer la façon dont les éléments recouverts sont affichés :

- `invisible` ou `transparent` pour les cacher ou les afficher en transparence ;
- `dynamic` ou `highly dynamic` pour les afficher en dégradé en fonction de la distance, ou pour accentuer le dégradé.

Code



Fichier Beamer3/code181.tex

```
%Exemple utilisant highly dynamic
\begin{itemize}[<+>-]
  \item Premier \item Deuxième \item
    Troisième
\end{itemize}
\begin{columns}
  \begin{column}{0.45\textwidth}
    \begin{exampleblock}<+>{Premier
      bloc}\end{exampleblock}
  \end{column}
  \begin{column}{0.45\textwidth}
    \begin{alertblock}<+>{Second
      bloc}\end{alertblock}
  \end{column}
\end{columns}
```

Sortie

- Premier
- Deuxième
- Troisième



Animation et superposition

Effets de transparence

`\setbeamercovered{mode}` permet de changer la façon dont les éléments recouverts sont affichés :

- `invisible` ou `transparent` pour les cacher ou les afficher en transparence ;
- `dynamic` ou `highly dynamic` pour les afficher en dégradé en fonction de la distance, ou pour accentuer le dégradé.

Code



Fichier Beamer3/code183.tex

```
%Exemple utilisant highly dynamic
\begin{itemize}[<+>-]
  \item Premier \item Deuxième \item
    Troisième
\end{itemize}
\begin{columns}
  \begin{column}{0.45\textwidth}
    \begin{exampleblock}<+>{Premier
      bloc}\end{exampleblock}
  \end{column}
  \begin{column}{0.45\textwidth}
    \begin{alertblock}<+>{Second
      bloc}\end{alertblock}
  \end{column}
\end{columns}
```

Sortie

- Premier
- Deuxième
- Troisième

Premier bloc

Second bloc

Animation et superposition

Effets de transparence

`\setbeamercovered{mode}` permet de changer la façon dont les éléments recouverts sont affichés :

- `invisible` ou `transparent` pour les cacher ou les afficher en transparence ;
- `dynamic` ou `highly dynamic` pour les afficher en dégradé en fonction de la distance, ou pour accentuer le dégradé.

Code



Fichier Beamer3/code185.tex

```
%Exemple utilisant highly dynamic
\begin{itemize}[<+>]
  \item Premier \item Deuxième \item
    Troisième
\end{itemize}
\begin{columns}
  \begin{column}{0.45\textwidth}
    \begin{exampleblock}<+>{Premier
      bloc}\end{exampleblock}
  \end{column}
  \begin{column}{0.45\textwidth}
    \begin{alertblock}<+>{Second
      bloc}\end{alertblock}
  \end{column}
\end{columns}
```

Sortie

- Premier
- Deuxième
- Troisième

Premier bloc

Second bloc

Animation et superposition

Effets de transparence

`\setbeamercovered{mode}` permet de changer la façon dont les éléments recouverts sont affichés :

- `invisible` ou `transparent` pour les cacher ou les afficher en transparence ;
- `dynamic` ou `highly dynamic` pour les afficher en dégradé en fonction de la distance, ou pour accentuer le dégradé.

Code



Fichier Beamer3/code187.tex

```
%Exemple utilisant highly dynamic
\begin{itemize}[<+>-]
  \item Premier \item Deuxième \item
    Troisième
\end{itemize}
\begin{columns}
  \begin{column}{0.45\textwidth}
    \begin{exampleblock}<+>{Premier
      bloc}\end{exampleblock}
  \end{column}
  \begin{column}{0.45\textwidth}
    \begin{alertblock}<+>{Second
      bloc}\end{alertblock}
  \end{column}
\end{columns}
```

Sortie

- Premier
- Deuxième
- Troisième

Premier bloc

Second bloc

Animation et superposition

Effets de transparence

`\setbeamercovered{mode}` permet de changer la façon dont les éléments recouverts sont affichés :

- `invisible` ou `transparent` pour les cacher ou les afficher en transparence ;
- `dynamic` ou `highly dynamic` pour les afficher en dégradé en fonction de la distance, ou pour accentuer le dégradé.

Code



Fichier Beamer3/code189.tex

```
%Exemple utilisant highly dynamic
\begin{itemize} [ <+>-> ]
  \item Premier \item Deuxième \item
    Troisième
\end{itemize}
\begin{columns}
  \begin{column} {0.45\textwidth}
    \begin{exampleblock} <+>-> {Premier
      bloc} \end{exampleblock}
  \end{column}
  \begin{column} {0.45\textwidth}
    \begin{alertblock} <+>-> {Second
      bloc} \end{alertblock}
  \end{column}
\end{columns}
```

Sortie

- Premier
- Deuxième
- Troisième

Premier bloc

Second bloc

Animation et superposition

Commandes et environnements supplémentaires

- `\visible<pages>{texte}` fait la même chose que `\uncover` mais le texte caché n'est jamais visible, même en transparence.

Animation et superposition

Commandes et environnements supplémentaires

- `\visible<pages>{texte}` fait la même chose que `\uncover` mais le texte caché n'est jamais visible, même en transparence.
- `\invisible<pages>{texte}` fait le contraire ;

Animation et superposition

Commandes et environnements supplémentaires

- `\visible<pages>{texte}` fait la même chose que `\uncover` mais le texte caché n'est jamais visible, même en transparence.
- `\invisible<pages>{texte}` fait le contraire ;
- `\begin{overlayarea}{largeur}{hauteur}` permet de faire une boîte de taille fixée dont le contenu peut varier :

Code



Fichier Beamer3/code207.tex

```
\begin{column}{0.5\textwidth}
  \only<1>{Un paragraphe .} \only<2>{Un
    paragraphe plus long.} \hrule
  Texte qui bouge
\end{column}
\begin{column}{0.5\textwidth}
  \begin{overlayarea}{\textwidth}{6ex}
  \only<1>{Un paragraphe .} \only<2>{
    Un paragraphe plus long.}
  \end{overlayarea} \hrule
  Texte qui ne bouge pas
\end{column}
```

Sortie

Un paragraphe. Texte qui bouge	Un paragraphe. Texte qui ne bouge pas
-----------------------------------	---

Animation et superposition

Commandes et environnements supplémentaires

- `\visible<pages>{texte}` fait la même chose que `\uncover` mais le texte caché n'est jamais visible, même en transparence.
- `\invisible<pages>{texte}` fait le contraire ;
- `\begin{overlayarea}{largeur}{hauteur}` permet de faire une boîte de taille fixée dont le contenu peut varier :

Code



Fichier Beamer3/code215.tex

```
\begin{column}{0.5\textwidth}
  \only<1>{Un paragraphe .}\only<2>{Un
    paragraphe plus long.}\hrule
  Texte qui bouge
\end{column}
\begin{column}{0.5\textwidth}
  \begin{overlayarea}{\textwidth}{6ex}
  \only<1>{Un paragraphe .}\only<2>{
    Un paragraphe plus long.}
  \end{overlayarea}\hrule
  Texte qui ne bouge pas
\end{column}
```

Sortie

Un paragraphe plus long. Texte qui bouge	Un paragraphe plus long. Texte qui ne bouge pas
--	--

Animation et superposition

Commandes et environnements supplémentaires

- `\visible<pages>{texte}` fait la même chose que `\uncover` mais le texte caché n'est jamais visible, même en transparence.
- `\invisible<pages>{texte}` fait le contraire ;
- `\begin{overlayarea}{largeur}{hauteur}` permet de faire une boîte de taille fixée dont le contenu peut varier ;
- l'environnement `onlyenv` fonctionne comme `\only` (et permet par exemple d'utiliser du `verbatim`) :

Code



Fichier Beamer3/code225.tex

```
\begin{onlyenv}<2>
\begin{verbatim*}
Saviez-vous que l'environnement
verbatim* fonctionne comme verbatim,
mais affiche les espaces du texte?
\end{verbatim*}
\end{onlyenv}
```

Sortie



Animation et superposition

Commandes et environnements supplémentaires

- `\visible<pages>{texte}` fait la même chose que `\uncover` mais le texte caché n'est jamais visible, même en transparence.
- `\invisible<pages>{texte}` fait le contraire ;
- `\begin{overlayarea}{largeur}{hauteur}` permet de faire une boîte de taille fixée dont le contenu peut varier ;
- l'environnement `onlyenv` fonctionne comme `\only` (et permet par exemple d'utiliser du `verbatim`) :

Code



Fichier Beamer3/code233.tex

```
\begin{onlyenv}<2>
\begin{verbatim*}
Saviez-vous que l'environnement
verbatim* fonctionne comme verbatim,
mais affiche les espaces du texte?
\end{verbatim*}
\end{onlyenv}
```

Sortie

Saviez-vous que l'environnement
verbatim* fonctionne comme verbatim,
mais affiche les espaces du texte?

Animation et superposition

Commandes et environnements supplémentaires

- `\visible<pages>{texte}` fait la même chose que `\uncover` mais le texte caché n'est jamais visible, même en transparence.
- `\invisible<pages>{texte}` fait le contraire ;
- `\begin{overlayarea}{largeur}{hauteur}` permet de faire une boîte de taille fixée dont le contenu peut varier ;
- l'environnement `onlyenv` fonctionne comme `\only` (et permet par exemple d'utiliser du verbatim) ;
- `\alt<pages>{alt1}{alt2}` affichera `alt1` sur les pages spécifiées, sinon `alt2` :

Code



Fichier Beamer3/code243.tex

```
Nous sommes sur la \alt<1>{première}{seconde} page.
```

Sortie

Nous sommes sur la première page.

Animation et superposition

Commandes et environnements supplémentaires

- `\visible<pages>{texte}` fait la même chose que `\uncover` mais le texte caché n'est jamais visible, même en transparence.
- `\invisible<pages>{texte}` fait le contraire ;
- `\begin{overlayarea}{largeur}{hauteur}` permet de faire une boîte de taille fixée dont le contenu peut varier ;
- l'environnement `onlyenv` fonctionne comme `\only` (et permet par exemple d'utiliser du verbatim) ;
- `\alt<pages>{alt1}{alt2}` affichera `alt1` sur les pages spécifiées, sinon `alt2` :

Code



Fichier Beamer3/code251.tex

```
Nous sommes sur la \alt<1>{première}{seconde} page.
```

Sortie

Nous sommes sur la seconde page.

Animation et superposition

Commandes et environnements supplémentaires

- `\visible<pages>{texte}` fait la même chose que `\uncover` mais le texte caché n'est jamais visible, même en transparence.
- `\invisible<pages>{texte}` fait le contraire ;
- `\begin{overlayarea}{largeur}{hauteur}` permet de faire une boîte de taille fixée dont le contenu peut varier ;
- l'environnement `onlyenv` fonctionne comme `\only` (et permet par exemple d'utiliser du verbatim) ;
- `\alt<pages>{alt1}{alt2}` affichera `alt1` sur les pages spécifiées, sinon `alt2` ;
- `\begin{altenv}<pages>{avant}{après}{sinon avant}{sinon après}` permet d'entourer son contenu sélectivement sur les pages spécifiées :

Code



Fichier Beamer3/code261.tex

```
\begin{altenv}<1>{\begin{center}}{\end{center}}{\begin{flushright}}{\end{flushright}}
\verb|TEXTE| à déplacer
\end{altenv}
```

Sortie

TEXTE à déplacer

Animation et superposition

Commandes et environnements supplémentaires

- `\visible<pages>{texte}` fait la même chose que `\uncover` mais le texte caché n'est jamais visible, même en transparence.
- `\invisible<pages>{texte}` fait le contraire;
- `\begin{overlayarea}{largeur}{hauteur}` permet de faire une boîte de taille fixée dont le contenu peut varier;
- l'environnement `onlyenv` fonctionne comme `\only` (et permet par exemple d'utiliser du verbatim);
- `\alt<pages>{alt1}{alt2}` affichera `alt1` sur les pages spécifiées, sinon `alt2`;
- `\begin{altenv}<pages>{avant}{après}{sinon avant}{sinon après}` permet d'entourer son contenu sélectivement sur les pages spécifiées :

Code



Fichier Beamer3/code269.tex

```
\begin{altenv}<1>{\begin{center}}{\end{center}}{\begin{flushright}}{\end{flushright}}
\verb|TEXTE| à déplacer
\end{altenv}
```

Sortie

TEXTE à déplacer

Animation et superposition

Affichage dynamique de tableaux

Il faut d'abord utiliser l'extension `colortbl`.

- affichage différé des lignes :

Code



Fichier Beamer3/code271.tex

```
\begin{tabular}{c|c|c}  
uuA&B&C\\ \hline  
uu\pause  
uuD&E&F\\ \hline  
uu\pause  
uuG&H&I  
\end{tabular}
```

Sortie

A	B	C
---	---	---

Animation et superposition

Affichage dynamique de tableaux

Il faut d'abord utiliser l'extension `colortbl`.

- affichage différé des lignes :

Code



Fichier Beamer3/code276.tex

```
\begin{tabular}{c|c|c}  
uuA&B&C\\ \hline  
uu\pause  
uuD&E&F\\ \hline  
uu\pause  
uuG&H&I  
\end{tabular}
```

Sortie

A	B	C
D	E	F

Animation et superposition

Affichage dynamique de tableaux

Il faut d'abord utiliser l'extension `colortbl`.

- affichage différé des lignes :

Code



Fichier Beamer3/code281.tex

```
\begin{tabular}{c|c|c}  
uuA&B&C\\ \hline  
uu\pause  
uuD&E&F\\ \hline  
uu\pause  
uuG&H&I  
\end{tabular}
```

Sortie

A	B	C
D	E	F
G	H	I

Animation et superposition

Affichage dynamique de tableaux

Il faut d'abord utiliser l'extension `colortbl`.

- affichage différé des lignes ;
- affichage différé des colonnes :

Code



Fichier Beamer3/code287.tex

```
\begin{tabular}{>{\onslide}c|>{\onslide<2->}c|>{\onslide<3>}c<{\onslide}}  
uuA&B&C\\\hline  
uuD&E&F\\\hline  
uuG&H&I  
\end{tabular}
```

Sortie

A		
D		
G		

Animation et superposition

Affichage dynamique de tableaux

Il faut d'abord utiliser l'extension `colortbl`.

- affichage différé des lignes ;
- affichage différé des colonnes :

Code



Fichier Beamer3/code292.tex

```
\begin{tabular}{>{\onslide}c|>{\onslide<2->}c|>{\onslide<3>}c<{\onslide}}
uuA&B&C\\ \hline
uuD&E&F\\ \hline
uuG&H&I
\end{tabular}
```

Sortie

A	B	
D	E	
G	H	

Animation et superposition

Affichage dynamique de tableaux

Il faut d'abord utiliser l'extension `colortbl`.

- affichage différé des lignes ;
- affichage différé des colonnes :

Code



Fichier Beamer3/code297.tex

```
\begin{tabular}{>{\onslide}c|>{\onslide<2->}c|>{\onslide<3>}c<{\onslide}}
uuA&B&C\\ \hline
uuD&E&F\\ \hline
uuG&H&I
\end{tabular}
```

Sortie

A	B	C
D	E	F
G	H	I

Animation et superposition

Affichage dynamique de tableaux

Il faut d'abord utiliser l'extension `colortbl`.

- affichage différé des lignes ;
- affichage différé des colonnes ;
- affichage différé des cellules :

Code



Fichier Beamer3/code304.tex

```
\begin{tabular}{|c|c|}  
  \hline  
  A&\uncover<2->{B}\\\hline  
  \uncover<3->{C}&\uncover<4>{D}\\\hline  
\end{tabular}
```

Sortie

A	

Animation et superposition

Affichage dynamique de tableaux

Il faut d'abord utiliser l'extension `colortbl`.

- affichage différé des lignes ;
- affichage différé des colonnes ;
- affichage différé des cellules :

Code



Fichier Beamer3/code309.tex

```
\begin{tabular}{|c|c|}  
  \hline  
  A&\uncover<2->{B}\\\hline  
  \uncover<3->{C}&\uncover<4>{D}\\\hline  
\end{tabular}
```

Sortie

A	B

Animation et superposition

Affichage dynamique de tableaux

Il faut d'abord utiliser l'extension `colortbl`.

- affichage différé des lignes ;
- affichage différé des colonnes ;
- affichage différé des cellules :

Code



Fichier Beamer3/code314.tex

```
\begin{tabular}{|c|c|}  
  \hline  
  A&\uncover<2->{B}\\\hline  
  \uncover<3->{C}&\uncover<4>{D}\\\hline  
\end{tabular}
```

Sortie

A	B
C	

Animation et superposition

Affichage dynamique de tableaux

Il faut d'abord utiliser l'extension `colortbl`.

- affichage différé des lignes ;
- affichage différé des colonnes ;
- affichage différé des cellules :

Code



Fichier Beamer3/code319.tex

```
\begin{tabular}{|c|c|}  
  \hline  
  A&\uncover<2->{B}\\\hline  
  \uncover<3->{C}&\uncover<4>{D}\\\hline  
\end{tabular}
```

Sortie

A	B
C	D

Options de l'environnement frame

```
\begin{frame}<pages>[<mode>] [options]{title}{subtitle}
```

- <pages> pour n'afficher que certaines pages du frame ;

Options de l'environnement frame

```
\begin{frame}<pages>[<mode>] [options]{title}{subtitle}
```

- <pages> pour n'afficher que certaines pages du frame ;
- [<mode>] pour un mode par défaut des sous-éléments (par exemple [<+>]);

Options de l'environnement frame

```
\begin{frame}<pages>[<mode>] [options]{title}{subtitle}
```

- <pages> pour n'afficher que certaines pages du frame ;
- [<mode>] pour un mode par défaut des sous-éléments (par exemple [<+>]);
- les [options] peuvent être :
 - allowframebreaks pour que le frame soit découpé si le contenu dépasse (à réserver pour la bibliographie !),
 - fragile pour inclure du verbatim ou des listings,
 - containsverbatim lorsque fragile ne suffit pas,
 - label=étiquette pour définir une étiquette (permet de réafficher le frame avec \againframe<pages>{étiquette}),
 - plain pour enlever les décorations,
 - shrink pour diminuer la taille de la fonte s'il y a trop de texte,
 - squeeze pour resserrer les espacements verticaux et rendre le frame plus compact.

Sectionnement

Il est possible (et même recommandé!) de structurer le document à l'extérieur des frames avec `\part`, `\section`, `\subsection`.

- `\partpage` pour faire une page de titre de partie (sur le modèle de `\titlepage`);
- `\tableofcontents[options]` pour afficher la table des matières.

Code



Fichier Beamer3/code321.tex

```
\begin{block}{Table des matières}
  \tableofcontents[pausesections,
    hideallsubsections]
\end{block}
```

Sortie

Table des matières

- 1 Types de documents supplémentaires en L^AT_EX

Sectionnement

Il est possible (et même recommandé!) de structurer le document à l'extérieur des frames avec `\part`, `\section`, `\subsection`.

- `\partpage` pour faire une page de titre de partie (sur le modèle de `\titlepage`);
- `\tableofcontents[options]` pour afficher la table des matières.

Code



Fichier Beamer3/code323.tex

```
\begin{block}{Table des matières}
  \tableofcontents[pausesections,
    hideallsubsections]
\end{block}
```

Sortie

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web

Sectionnement

Il est possible (et même recommandé!) de structurer le document à l'extérieur des frames avec `\part`, `\section`, `\subsection`.

- `\partpage` pour faire une page de titre de partie (sur le modèle de `\titlepage`);
- `\tableofcontents[options]` pour afficher la table des matières.

Code



Fichier Beamer3/code325.tex

```
\begin{block}{Table des matières}
\tableofcontents[pausesections,
hideallsubsections]
\end{block}
```

Sortie

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés

Sectionnement

Il est possible (et même recommandé!) de structurer le document à l'extérieur des frames avec `\part`, `\section`, `\subsection`.

- `\partpage` pour faire une page de titre de partie (sur le modèle de `\titlepage`);
- `\tableofcontents[options]` pour afficher la table des matières.

Code



Fichier Beamer3/code327.tex

```
\begin{block}{Table des matières}
  \tableofcontents[pausesections,
    hideallsubsections]
\end{block}
```

Sortie

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Sectionnement

Options de la table des matières

- `pausesections` ou `pausesubsections` pour faire une pause à chaque élément ;

Sectionnement

Options de la table des matières

- `pausesections` ou `pausesubsections` pour faire une pause à chaque élément ;
- `sectionstyle=courante/autres` pour changer le style d'affichage de la section courante et des autres : `show`, `shaded` ou `hide` ;
- `subsectionstyle=courante/soeurs/cousines` pour changer le style d'affichage de la sous-section courante, de ses soeurs (filles de la même section) et de ses cousines (filles des autres sections) ;

Sectionnement

Options de la table des matières

- `pausesections` ou `pausesubsections` pour faire une pause à chaque élément ;
- `sectionstyle=courante/autres` pour changer le style d'affichage de la section courante et des autres : `show`, `shaded` ou `hide` ;
- `subsectionstyle=courante/soeurs/cousines` pour changer le style d'affichage de la sous-section courante, de ses soeurs (filles de la même section) et de ses cousines (filles des autres sections) ;
- `currentsection` est équivalent à `sectionstyle=show/shaded`, `subsectionstyle=show/show/shaded` ;
- `currentsubsection` est équivalent à `subsectionstyle=show/shaded` ;

Sectionnement

Options de la table des matières

- `pausesections` ou `pausesubsections` pour faire une pause à chaque élément ;
- `sectionstyle=courante/autres` pour changer le style d'affichage de la section courante et des autres : `show`, `shaded` ou `hide` ;
- `subsectionstyle=courante/soeurs/cousines` pour changer le style d'affichage de la sous-section courante, de ses soeurs (filles de la même section) et de ses cousines (filles des autres sections) ;
- `currentsection` est équivalent à `sectionstyle=show/shaded`, `subsectionstyle=show/show/shaded` ;
- `currentsubsection` est équivalent à `subsectionstyle=show/shaded` ;
- `hideallsubsections` est équivalent à `subsectionstyle=hide` ;
- `hideothersubsections` est équivalent à `subsectionstyle=show/show/hide`.

Sommaires automatiques (à mettre dans le préambule)

Beamer3/code329.tex



Fichier

```
\AtBeginPart{
  \begin{frame}
    \partpage
  \end{frame}
}
```

```
\AtBeginSection{
  \begin{frame}
    \frametitle{Table des matières}
    \tableofcontents[sectionstyle=show/shaded,subsectionstyle=show/
      shaded/hide]
  \end{frame}
}
```

```
\AtBeginSubsection{
  \begin{frame}
    \frametitle{Table des matières}
    \tableofcontents[sectionstyle=show/shaded,subsectionstyle=show/
      shaded/hide]
  \end{frame}
}
```

Modes de Beamer

Le mode est spécifié en option de la classe.

- `beamer` pour la version vidéoprojecteur ;
- `handout` pour la version papier à distribuer pendant la présentation ;
- `trans` pour faire des transparents pour rétroprojecteur ;
- `article` pour une version sous forme d'article papier.

Modes de Beamer

Le mode `handout`

Il est recommandé d'utiliser un thème clair.

On peut utiliser l'extension `pgfpages` pour disposer plusieurs pages par feuille.

Modes de Beamer

Le mode handout

Il est recommandé d'utiliser un thème clair.

On peut utiliser l'extension `pgfpages` pour disposer plusieurs pages par feuille.

Exemple de configuration



Fichier Beamer3/code331.tex

```
\documentclass[french,handout]{beamer}

\usepackage{pgfpages}
%au_choix:
\pgfpagesuselayout{2_on_1}[a4paper]
\pgfpagesuselayout{4_on_1}[a4paper,landscape]

\begin{document}
...
```

Modes de Beamer

Le mode `trans`

Il est recommandé d'utiliser un thème « blanc ».

Modes de Beamer

Le mode trans

Il est recommandé d'utiliser un thème « blanc ».

Exemple de configuration



Fichier Beamer3/code333.tex

```
\documentclass[french,trans]{beamer}

\begin{document}
...
```

Modes de Beamer

Le mode `article`

Il est préférable d'utiliser la classe `article` et l'extension `beamerarticle`.
Les thèmes, les blocs et les effets de superposition seront désactivés.

Modes de Beamer

Le mode `article`

Il est préférable d'utiliser la classe `article` et l'extension `beamerarticle`.

Les thèmes, les blocs et les effets de superposition seront désactivés.

Exemple de configuration



Fichier Beamer3/code335.tex

```
\documentclass[french]{article}

\RequirePackage{beamerarticle}

\begin{document}
...
```

Modes de Beamer

Adapter le contenu au mode

- `\mode<liste>{texte}` lira le `texte` uniquement si le mode courant est dans la `liste`;

Modes de Beamer

Adapter le contenu au mode

- `\mode<liste>{texte}` lira le texte uniquement si le mode courant est dans la liste;
- ça marche aussi avec `\item`, `\only...`

Modes de Beamer

Adapter le contenu au mode

- `\mode<liste>{texte}` lira le texte uniquement si le mode courant est dans la liste;
- ça marche aussi avec `\item`, `\only...`
- `\mode`
`<liste>`
lira ce qui suit uniquement si le mode courant est dans la liste.

Ne pas oublier de le désactiver avec `\mode`
`<all>`

Le mode "all" représente n'importe quel mode, et "presentation" n'importe quel mode à part article.

Modes de Beamer

Adapter le contenu au mode

- `\mode<liste>{texte}` lira le texte uniquement si le mode courant est dans la liste;
- ça marche aussi avec `\item`, `\only...`
- `\mode`
`<liste>`
lira ce qui suit uniquement si le mode courant est dans la liste.

Ne pas oublier de le désactiver avec `\mode`
`<all>`

Le mode "all" représente n'importe quel mode, et "presentation" n'importe quel mode à part article.

Code



Fichier Beamer3/code339.tex

```
Les modes suivants sont activés :  
\begin{itemize}  
  \item<presentation>{presentation}  
  \item<beamer>{beamer}  
  \item<handout>{handout}  
  \item<trans>{trans}  
  \item<article>{article}  
\end{itemize}
```

Sortie

Les modes suivants sont activés :

- presentation
- beamer

Exercice

Mise en application

- 1 Écrire un document Beamer avec quelques frames.

Exercice

Mise en application

- 1 Écrire un document Beamer avec quelques frames.
- 2 Y ajouter des colonnes et des blocs.

Exercice

Mise en application

- 1 Écrire un document Beamer avec quelques frames.
- 2 Y ajouter des colonnes et des blocs.
- 3 Faire quelques effets d'affichage différé ou de superposition.

Exercice

Mise en application

- 1 Écrire un document Beamer avec quelques frames.
- 2 Y ajouter des colonnes et des blocs.
- 3 Faire quelques effets d'affichage différé ou de superposition.
- 4 Sectionner le document et afficher des tables des matières.

Exercice

Mise en application

- 1 Écrire un document Beamer avec quelques frames.
- 2 Y ajouter des colonnes et des blocs.
- 3 Faire quelques effets d'affichage différé ou de superposition.
- 4 Sectionner le document et afficher des tables des matières.
- 5 Faire une version pour rétroprojecteur, ou à distribuer au public.

Exercice

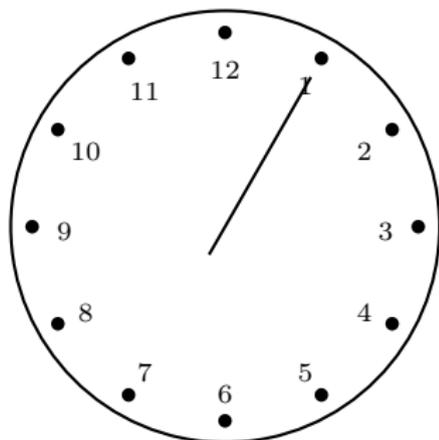
Mise en application

- 1 Écrire un document Beamer avec quelques frames.
- 2 Y ajouter des colonnes et des blocs.
- 3 Faire quelques effets d'affichage différé ou de superposition.
- 4 Sectionner le document et afficher des tables des matières.
- 5 Faire une version pour rétroprojecteur, ou à distribuer au public.
- 6 Convertir le document en un article.

Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

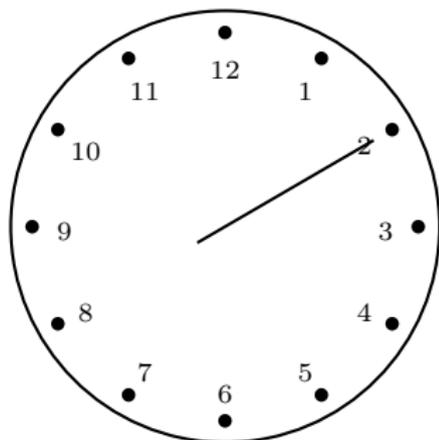
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

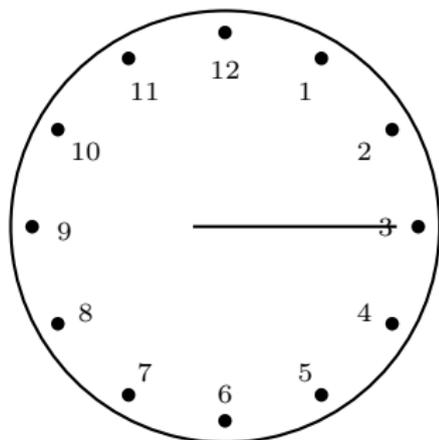
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

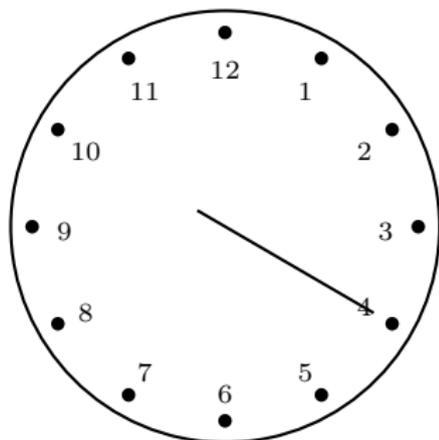
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

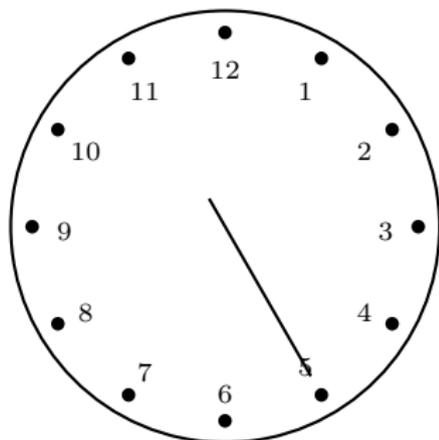
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

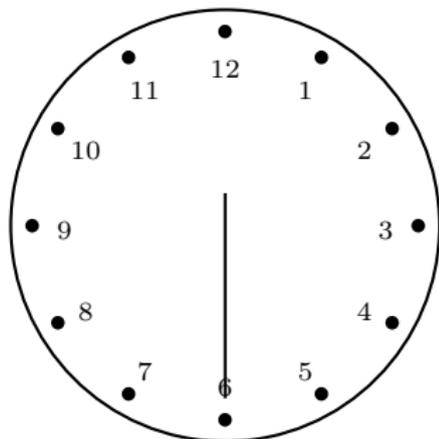
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

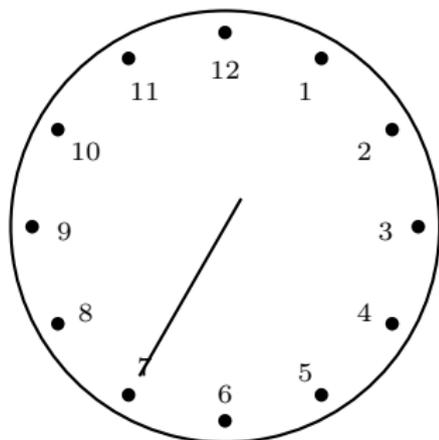
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

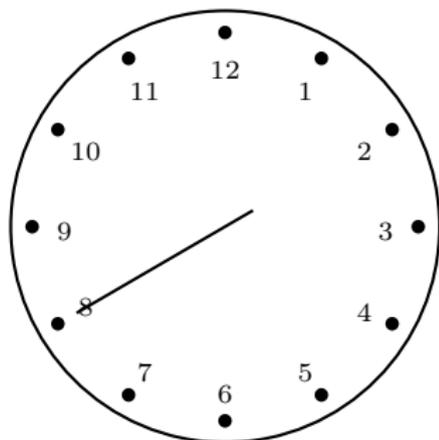
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

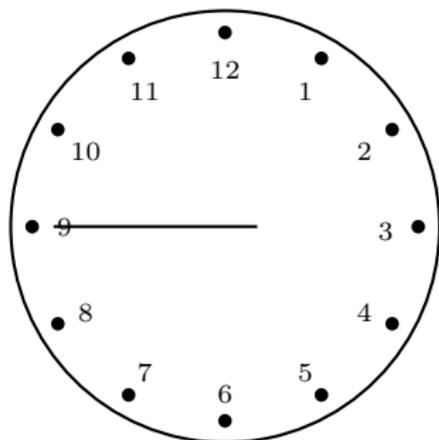
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

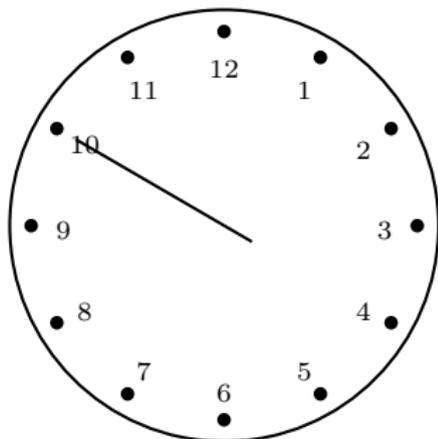
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

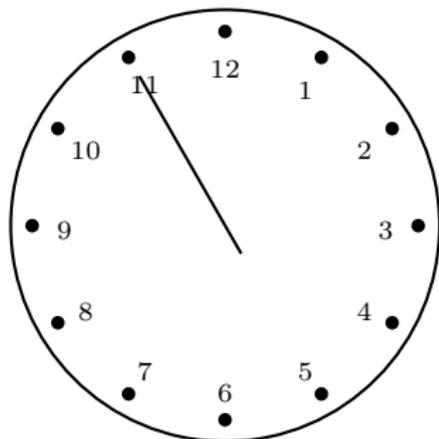
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

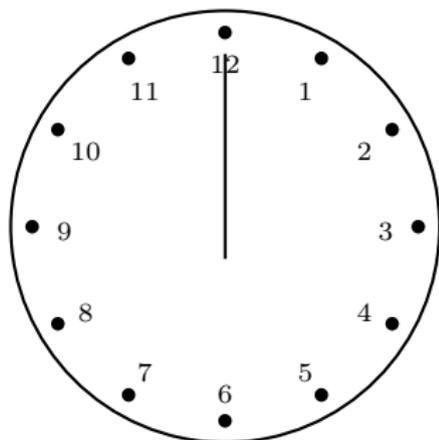
Sortie



Exercice [Difficile]

Trouver le code pour obtenir la sortie suivante

Sortie



Code



Fichier Beamer3/code340.tex

```
\transduration <1-11>{1}
\SpecialCoor
\degrees [12]
\begin{pspicture}(-5,-3)(5,3)
uu\pscircle{2}
uu\multido{\nn=1+1}{12}{
uuuu\rput{3}{
uuuuuu\rput{-3}(1.5;-\nn){\nn}
uuuuuu\only<\nn>{
uuuuuuuu\psline(-0.3;-\nn)(1.6;-\nn)
uuuuuu}
uuuu}
uuuu\psdots(1.8;\nn)
uu}
\end{pspicture}
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
 - Prérequis
 - Conversions
 - HeVeA et HachA
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
 - **Prérequis**
 - Conversions
 - HeVeA et HachA
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Comment disposer d'un site web ?

La plupart des facultés proposent un service d'hébergement de pages personnelles.

Comment disposer d'un site web ?

La plupart des facultés proposent un service d'hébergement de pages personnelles.

Une fois activé, le compte dispose d'un répertoire (par exemple `wwwdoc`) dont l'arborescence est visible depuis l'extérieur à une adresse racine (par exemple `www.mondépartement.mafac.fr/username/`).

Comment disposer d'un site web ?

La plupart des facultés proposent un service d'hébergement de pages personnelles.

Une fois activé, le compte dispose d'un répertoire (par exemple `wwwdoc`) dont l'arborescence est visible depuis l'extérieur à une adresse racine (par exemple `www.mondépartement.mafac.fr/username/`).

La page web par défaut d'un répertoire doit s'appeler (en général) `index.html`.

Comment disposer d'un site web ?

La plupart des facultés proposent un service d'hébergement de pages personnelles.

Une fois activé, le compte dispose d'un répertoire (par exemple `wwwdoc`) dont l'arborescence est visible depuis l'extérieur à une adresse racine (par exemple `www.mondépartement.mafac.fr/username/`).

La page web par défaut d'un répertoire doit s'appeler (en général) `index.html`.

On peut envoyer des fichiers à distance par FTP ou SCP.

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
 - Prérequis
 - **Conversions**
 - HeVeA et HachA
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Convertir du HTML vers \LaTeX

On peut utiliser `html2latex` :

`http://htmltolatex.sourceforge.net/`

Son avantage est d'être écrit en Java (donc multiplateforme).

Convertir du \LaTeX en HTML

Il existe plusieurs solutions :

- `latex2html` (formules rendues en images, nécessite Perl) ;
- `TeX4ht` (formules rendues en images, nécessite ImageMagick) ;
- `TtH` (formules rendues en HTML) ;
- `HeVeA` (formules rendues au choix en images ou en HTML) ;

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
 - Prérequis
 - Conversions
 - HeVeA et HachA
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée

Fonctionnement

Dans un premier temps, on compile avec HeVeA :

Usage

```
hevea monfichier.tex
```

Fonctionnement

Dans un premier temps, on compile avec HeVeA :

Usage

```
hevea monfichier.tex
```

Puis, HachA permet de découper le fichier HTML obtenu en petits morceaux avec des liens de navigation en suivant la hiérarchie du document.

Usage

```
hacha monfichier.html -o index.html
```

On peut utiliser l'environnement `toimage` :

Exemple



Fichier Beamer3/code345.tex

```
\begin{toimage}  
□□$$\sum_{k=1}^+\infty\frac{1}{k^2}=\frac{\pi^2}{6}$$  
\end{toimage}
```

Formules en images

On peut utiliser l'environnement `toimage` :

Exemple



Fichier Beamer3/code346.tex

```
\begin{toimage}
□□$$\sum_{k=1}^+\infty\frac{1}{k^2}=\frac{\pi^2}{6}$$
\end{toimage}
```

Un fichier externe contenant les formules est généré, puis compilé et converti en une série d'images GIF à l'aide d'un script fourni avec HeVeA : `imagen`.

Commandes HeVeA propres au HTML

- L'environnement `rawhtml` permet de taper du code HTML directement (un peu comme `verbatim`);
- `\newstyle{élément}{styles}` permet d'exporter le formatage d'un élément dans le fichier CSS;
- `\begin{divstyle}{classe CSS}` permet de créer un élément HTML `<DIV>` de la classe spécifiée;
- `\@open{TAG}{attributs}... \@close{TAG}` pour ouvrir et fermer un élément HTML;
- `\ahref{lien}{texte}` pour faire un lien.

La classe site

Elle permet de faire un site avec un menu et un style de couleur configurable. Chaque commande de sectionnement ajoute une nouvelle page au site.

Exemple



Fichier Beamer3/code347.tex

```
\documentclass{site}

\usecolortheme{blue}

\title[Accueil de mon site]{Accueil}

\begin{document}
...
\end{document}
```

Exercice

Faire un essai avec HeVeA, et éventuellement avec la classe `site`

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
 - Les fontes
 - Calculer en \LaTeX
 - En finir avec les **bad boxes**
 - Enlever les espaces indésirables
 - Écrire dans des fichiers
 - Divers
- 4 Programmation \LaTeX avancée

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
 - Les fontes
 - Calculer en \LaTeX
 - En finir avec les **bad boxes**
 - Enlever les espaces indésirables
 - Écrire dans des fichiers
 - Divers
- 4 Programmation \LaTeX avancée

Les attributs de fontes

Il y en a cinq en tout :

- 1 l'encodage (encoding) qui peut être :
 - OT1 pour l'ancien format de texte ;
 - T1 pour le nouveau format de texte ;
 - OML pour les maths en italique ;
 - OMS pour les symboles mathématiques ;
 - OMX pour les gros symboles ;
 - U pour "inconnu".

Les attributs de fontes

Il y en a cinq en tout :

- ① l'encodage (encoding) .
- ② la famille (family) qui peut être :
 - `cmr` pour "Computer Modern Roman" ;
 - `cmss` pour "Computer Modern Sans" ;
 - `cmtt` pour "Computer Modern Typewriter" ;
 - `cmm` pour "Computer Modern Math Italic" ;
 - `cmsy` pour "Computer Modern Math Symbols" ;
 - `cmsy` pour "Computer Modern Math Extensions" ;
 - `ptm` pour "Adobe Times" ;
 - `phv` pour "Adobe Helvetica" ;
 - `pcr` pour "Adobe Courier".
 - ...

Les attributs de fontes

Il y en a cinq en tout :

- ① l'encodage (encoding) .
- ② la famille (family) .
- ③ les séries de graisse (series) qui peuvent être :
 - m pour "Medium" ;
 - b pour "Bold" ;
 - bx pour "Bold extended" ;
 - sb pour "Semi-bold" ;
 - c pour "Condensed" .

Les attributs de fontes

Il y en a cinq en tout :

- ① l'encodage (encoding) .
- ② la famille (family) .
- ③ les séries de graisse (series) .
- ④ la forme (shape) qui peut être :
 - `n` pour "Normal" ;
 - `it` pour "Italic" ;
 - `sl` pour "Slanted" ;
 - `sc` pour "Small caps".

Les attributs de fontes

Il y en a cinq en tout :

- 1 l'encodage (encoding) .
- 2 la famille (family) .
- 3 les séries de graisse (series) .
- 4 la forme (shape) .
- 5 la taille (size), généralement exprimée en points.

Les attributs de fontes

Techniques usuelles pour changer les attributs

Les valeurs indiquées sont celles par défaut de la classe `article`.

- 1 l'encodage (par défaut `OT1`) se change avec l'extension `fontenc` (préférer `T1`).

Les attributs de fontes

Techniques usuelles pour changer les attributs

Les valeurs indiquées sont celles par défaut de la classe `article`.

- 1 l'encodage (par défaut `OT1`) se change avec l'extension `fontenc` (préférer `T1`).
- 2 la famille se change avec `\text{...}{...}` ou `\...family` :
 - `\textrm` ou `\rmfamily` pour `cmr` ;
 - `\textsf` ou `\sffamily` pour `cmss` ;
 - `\texttt` ou `\ttfamily` pour `cmtt`.

Les attributs de fontes

Techniques usuelles pour changer les attributs

Les valeurs indiquées sont celles par défaut de la classe `article`.

- ❶ l'encodage (par défaut OT1) se change avec l'extension `fontenc` (préférer T1).
- ❷ la famille se change avec `\text{...}{...}` ou `\...family`.
- ❸ les séries se changent avec `\text{...}{...}` ou `\...series` :
 - `\textmd` ou `\mdseries` pour m;
 - `\textbf` ou `\bfseries` pour bx.

Les attributs de fontes

Techniques usuelles pour changer les attributs

Les valeurs indiquées sont celles par défaut de la classe `article`.

- ❶ l'encodage (par défaut `OT1`) se change avec l'extension `fontenc` (préférer `T1`).
- ❷ la famille se change avec `\text{...}{...}` ou `\...family`.
- ❸ les séries se changent avec `\text{...}{...}` ou `\...series`.
- ❹ les formes se changent avec `\text{...}{...}` ou `\...shape` :
 - `\textup` ou `\upshape` pour `n` ;
 - `\textit` ou `\itshape` pour `it` ;
 - `\textsl` ou `\slshape` pour `sl` ;
 - `\textsc` ou `\scshape` pour `sc`.

Les attributs de fontes

Techniques usuelles pour changer les attributs

Les valeurs indiquées sont celles par défaut de la classe `article`.

- ❶ l'encodage (par défaut `OT1`) se change avec l'extension `fontenc` (préférer `T1`).
- ❷ la famille se change avec `\text...{...}` ou `\...family`.
- ❸ les séries se changent avec `\text...{...}` ou `\...series`.
- ❹ les formes se changent avec `\text...{...}` ou `\...shape`.
- ❺ les tailles se changent avec :
 - `\tiny` pour 5pt ;
 - `\scriptsize` pour 7pt ;
 - `\footnotesize` pour 8pt ;
 - `\small` pour 9pt ;
 - `\normalsize` pour 10pt ;
 - `\large` pour 12pt ;
 - `\Large` pour 14.4pt ;
 - `\LARGE` pour 17.28pt ;
 - `\huge` pour 20.74pt ;
 - `\Huge` pour 24.88pt.

Les attributs de fontes

Modification locale des attributs

- `\fontencoding{encoding}` pour changer l'encodage ;
- `\fontfamily{family}` pour changer la famille ;
- `\fontseries{series}` pour changer les séries ;
- `\fontshape{shape}` pour changer la forme ;

Les attributs de fontes

Modification locale des attributs

- `\fontencoding{encoding}` pour changer l'encodage ;
- `\fontfamily{family}` pour changer la famille ;
- `\fontseries{series}` pour changer les séries ;
- `\fontshape{shape}` pour changer la forme ;
- `\fontsize{size}{baselineskip}` pour changer la taille ainsi que la hauteur de ligne ;

Les attributs de fontes

Modification locale des attributs

- `\fontencoding{encoding}` pour changer l'encodage ;
- `\fontfamily{family}` pour changer la famille ;
- `\fontseries{series}` pour changer les séries ;
- `\fontshape{shape}` pour changer la forme ;
- `\fontsize{size}{baselineskip}` pour changer la taille ainsi que la hauteur de ligne ;
- `\selectfont` active les changements précédemment sélectionnés.

Ne pas écrire de texte entre un changement d'attribut et `\selectfont` !

`\selectfont` ne s'applique qu'au groupe en cours ;

Les attributs de fontes

Modification locale des attributs

- `\fontencoding{encoding}` pour changer l'encodage ;
- `\fontfamily{family}` pour changer la famille ;
- `\fontseries{series}` pour changer les séries ;
- `\fontshape{shape}` pour changer la forme ;
- `\fontsize{size}{baselineskip}` pour changer la taille ainsi que la hauteur de ligne ;
- `\selectfont` active les changements précédemment sélectionnés.

Ne pas écrire de texte entre un changement d'attribut et `\selectfont` !

`\selectfont` ne s'applique qu'au groupe en cours ;

- `\usefont{encoding}{family}{series}{shape}` appelle toutes ces commandes en une fois (sauf `\fontsize`).

Nouvelles commandes de sélection de fonte

À utiliser dans le préambule :

- `\DeclareFixedFont\cmd{enc}{family}{series}{shape}{size}`
déclare `\cmd` qui active la police sélectionnée, sur le modèle de `\itshape`,
`\bfseries...`

Nouvelles commandes de sélection de fonte

À utiliser dans le préambule :

- `\DeclareFixedFont\cmd{enc}{family}{series}{shape}{size}`
déclare `\cmd` qui active la police sélectionnée, sur le modèle de `\itshape`,
`\bfseries...`
- `\DeclareTextFontCommand\cmd{...}` déclare `\cmd` sur le modèle de
`\textit`, `\textbf...`

Par exemple, `\textrm` a été implémentée en utilisant
`\DeclareTextFontCommand{\textrm}{\rmfamily}`.

Nouvelles commandes de sélection de fonte

À utiliser dans le préambule :

- `\DeclareFixedFont\cmd{enc}{family}{series}{shape}{size}`
déclare `\cmd` qui active la police sélectionnée, sur le modèle de `\itshape`, `\bfseries`...
- `\DeclareTextFontCommand\cmd{...}` déclare `\cmd` sur le modèle de `\textit`, `\textbf`...

Par exemple, `\textrm` a été implémentée en utilisant `\DeclareTextFontCommand{\textrm}{\rmfamily}`.

Code



Fichier Beamer3/code352.tex

```
%Dans le préambule :  
\DeclareFixedFont\funnynormal{T1}{  
  cmfr}{m}{n}{10pt}  
\DeclareTextFontCommand\textfunny{\  
  funnynormal}  
  
Du texte en \textfunny{Computer  
  Modern funny roman}.
```

Sortie

Du texte en Computer Modern funny roman.

Quelques exemples de fontes répandues

Computer Modern roman, bold, normal (T1/cmr/b/n):

ABCDEFGHIJKL abcdefghijkl 0123456789

Computer Modern roman, bold extended, normal (T1/cmr/bx/n):

ABCDEFGHIJKL abcdefghijkl 0123456789

Helvetica, bold, normal (T1/phv/b/n):

ABCDEFGHIJKL abcdefghijkl 0123456789

Helvetica, bold condensed, normal (T1/phv/bc/n):

ABCDEFGHIJKL abcdefghijkl 0123456789

Times, bold, small capitals (T1/ptm/b/sc):

ABCDEFGHIJKL ABCDEFGHIJKL 0123456789

New Century, medium, normal (T1/pnc/m/n):

ABCDEFGHIJKL abcdefghijkl 0123456789

Les attributs de fontes

Modification globale des attributs

- La fonte à utiliser par défaut (pour `\normalfont`) peut être modifiée en redéfinissant les commandes suivantes :
 - `\encodingdefault` (par défaut `OT1`);
 - `\familydefault` (par défaut `\rmdefault`);
 - `\seriesdefault` (par défaut `\mddefault`);
 - `\shapedefault` (par défaut `\updefault`).

Les attributs de fontes

Modification globale des attributs

- La fonte à utiliser par défaut (pour `\normalfont`) peut être modifiée.
- Les familles à utiliser (pour `\rmfamily`, `\sffamily` ou `\ttfamily`) peuvent être modifiées en redéfinissant les commandes suivantes :
 - `\rmdefault` (par défaut `cmr`);
 - `\sfdefault` (par défaut `cmss`);
 - `\ttdefault` (par défaut `cmtt`).

Les attributs de fontes

Modification globale des attributs

- La fonte à utiliser par défaut (pour `\normalfont`) peut être modifiée.
- Les familles à utiliser (pour `\rmfamily`, `\sffamily` ou `\ttfamily`) peuvent être modifiées.
- Les séries à utiliser (pour `\bfseries` ou `\mdseries`) peuvent être modifiées en redéfinissant les commandes suivantes :
 - `\bfdefault` (par défaut `bx`);
 - `\mddefault` (par défaut `m`).

Les attributs de fontes

Modification globale des attributs

- La fonte à utiliser par défaut (pour `\normalfont`) peut être modifiée.
- Les familles à utiliser (pour `\rmfamily`, `\sffamily` ou `\ttfamily`) peuvent être modifiées.
- Les séries à utiliser (pour `\bfseries` ou `\mdseries`) peuvent être modifiées.
- Les formes à utiliser (pour `\itshape`, `\slshape`, `\scshape` ou `\upshape`) peuvent être modifiées en redéfinissant les commandes suivantes :
 - `\itdefault` (par défaut `it`);
 - `\sldefault` (par défaut `sl`);
 - `\scdefault` (par défaut `sc`);
 - `\updefault` (par défaut `n`).

Les attributs de fontes

Modification globale des attributs

- La fonte à utiliser par défaut (pour `\normalfont`) peut être modifiée.
- Les familles à utiliser (pour `\rmfamily`, `\sffamily` ou `\ttfamily`) peuvent être modifiées.
- Les séries à utiliser (pour `\bfseries` ou `\mdseries`) peuvent être modifiées.
- Les formes à utiliser (pour `\itshape`, `\slshape`, `\scshape` ou `\upshape`) peuvent être modifiées.

Code



Fichier Beamer3/code362.tex

```
\usefont{T1}{phv}{m}{n}
Voici la \rmfamily \_fonte d'origine.

\renewcommand\rmdefault{phv}
%On fait un changement de fonte bidon
\rmfamily
Voici la \rmfamily \_nouvelle fonte qui
reste active.
```

Sortie

Voici la fonte d'origine.
Voici la nouvelle fonte qui
reste active.

Changer les fontes de tout le document avec une extension

Polices par défaut

`\rmfamily: Computer Modern roman`, medium, normal (T1/cmr/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\sffamily: Computer Modern sans sérif`, medium, normal (T1/cmss/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\ttfamily: Computer Modern typewriter`, medium, normal (T1/cmtt/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\bfseries: Computer Modern roman`, bold, normal (T1/cmr/b/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\itshape: Computer Modern roman`, medium, italic (T1/cmr/m/it)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\slanted: Computer Modern roman`, medium, slanted (T1/cmr/m/sl)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\scshape: Computer Modern roman`, medium, small capitals (T1/cmr/m/sc)

ABCDEFGHJKLM ABCDEFGHIJKL 0123456789

Changer les fontes de tout le document avec une extension

L'extension `beton` (à utiliser avec `euler` pour les maths)

`\rmfamily: Concrete`, medium, normal (T1/ccr/m/n)

ABCDEFGHIJKLM abcdefghijkl 0123456789

`\sffamily: —`

Manquante

`\ttfamily: —`

Manquante

`\bfseries: —`

Manquante

`\itshape: Concrete`, medium, italic (T1/ccr/m/it)

ABCDEFGHIJKLM abcdefghijkl 0123456789

`\slanted: Concrete`, medium, slanted (T1/ccr/m/sl)

ABCDEFGHIJKLM abcdefghijkl 0123456789

`\scshape: Concrete`, medium, small capitals (T1/ccr/m/sc)

ABCDEFGHIJKLM ABCDEFGHIJKL 0123456789

Changer les fontes de tout le document avec une extension

L'extension bookman

`\rmfamily: Bookman`, medium, normal (T1/pbk/m/n)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\sffamily: Avant Garde`, medium, normal (T1/pag/m/n)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\ttfamily: Courier`, medium, normal (T1/pcr/m/n)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\bfseries: Bookman`, bold, normal (T1/pbk/b/n)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\itshape: Bookman`, medium, italic (T1/pbk/m/it)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\slanted: Bookman`, medium, slanted (T1/pbk/m/sl)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\scshape: Bookman`, medium, small capitals (T1/pbk/m/sc)

ABCDEFGHIJKL ABCDEFGHIJKL 0123456789

Changer les fontes de tout le document avec une extension

L'extension charter

`\rmfamily: Charter`, medium, normal (T1/bch/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\sffamily: Charter`, medium, normal (T1/bch/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\ttfamily: Charter`, medium, normal (T1/bch/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\bfseries: Charter`, bold, normal (T1/bch/b/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\itshape: Charter`, medium, italic (T1/bch/m/it)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\slanted: Charter`, medium, slanted (T1/bch/m/sl)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\scshape: Charter`, medium, small capitals (T1/bch/m/sc)

ABCDEFGHJKLM ABCDEFGHIJKL 0123456789

Changer les fontes de tout le document avec une extension

L'extension `lmodern`

`\rmfamily: Latin Modern roman`, medium, normal (T1/lmr/m/n)

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijkl 0123456789

`\sffamily: Latin Modern sans sérif`, medium, normal (T1/lmss/m/n)

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijkl 0123456789

`\ttfamily: Latin Modern typewriter`, medium, normal (T1/lmtt/m/n)

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijkl 0123456789

`\bfseries: Latin Modern roman`, bold, normal (T1/lmr/b/n)

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijkl 0123456789

`\itshape: Latin Modern roman`, medium, italic (T1/lmr/m/it)

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijkl 0123456789

`\slanted: Latin Modern roman`, medium, slanted (T1/lmr/m/sl)

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijkl 0123456789

`\scshape: Latin Modern roman`, medium, small capitals (T1/lmr/m/sc)

ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKL 0123456789

Changer les fontes de tout le document avec une extension

L'extension newcent

`\rmfamily: New Century`, medium, normal (T1/pnc/m/n)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\sffamily: Computer Modern sans sérif`, medium, normal (T1/cmss/m/n)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\ttfamily: Computer Modern typewriter`, medium, normal (T1/cmтт/m/n)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\bfseries: New Century`, bold, normal (T1/pnc/b/n)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\itshape: New Century`, medium, italic (T1/pnc/m/it)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\slanted: New Century`, medium, slanted (T1/pnc/m/sl)

ABCDEFGHIJKL abcdefghijkl 0123456789

`\scshape: New Century`, medium, small capitals (T1/pnc/m/sc)

ABCDEFGHIJKL ABCDEFGHIJKL 0123456789

Changer les fontes de tout le document avec une extension

L'extension palatino

`\rmfamily: Palatino`, medium, normal (T1/pp1/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\sffamily: Helvetica`, medium, normal (T1/phv/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\ttfamily: Courier`, medium, normal (T1/pcr/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\bfseries: Palatino`, bold, normal (T1/pp1/b/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\itshape: Palatino`, medium, italic (T1/pp1/m/it)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\slanted: Palatino`, medium, slanted (T1/pp1/m/sl)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\scshape: Palatino`, medium, small capitals (T1/pp1/m/sc)

ABCDEFGHJKLM ABCDEFGHIJKL 0123456789

Changer les fontes de tout le document avec une extension

L'extension times

`\rmfamily: Times`, medium, normal (T1/ptm/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\sffamily: Helvetica`, medium, normal (T1/phv/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\ttfamily: Courier`, medium, normal (T1/pcr/m/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\bfseries: Times`, bold, normal (T1/ptm/b/n)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\itshape: Times`, medium, italic (T1/ptm/m/it)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\slanted: Times`, medium, slanted (T1/ptm/m/sl)

ABCDEFGHJKLM abcdefghijkl 0123456789

`\scshape: Times`, medium, small capitals (T1/ptm/m/sc)

ABCDEFGHJKLM ABCDEFGHIJKL 0123456789

Exercice

Donner trois commandes équivalentes à l'utilisation de l'extension `times`

Solution

Donner trois commandes équivalentes à l'utilisation de l'extension `times`

Code



Fichier Beamer3/code364.tex

```
\renewcommand\rmdefault{ptm}  
\renewcommand\sfddefault{phv}  
\renewcommand\ttdefault{pcr}
```

Exercice

Définir une commande pour écrire en gothique

Solution

Définir une commande pour écrire en gothique

Code



Fichier Beamer3/code365.tex

```
\DeclareFixedFont\gotnormal{U}{yfrak}{m}{n}{10pt}  
\DeclareTextFontCommand\textgot{\gotnormal}
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
 - Les fontes
 - Calculer en \LaTeX
 - En finir avec les **bad boxes**
 - Enlever les espaces indésirables
 - Écrire dans des fichiers
 - Divers
- 4 Programmation \LaTeX avancée

L'extension calc

Redéfinit les commandes `\setcounter`, `\addtocounter`, `\setlength` et `\addtolength`, qui acceptent dès lors des expressions algébriques.

L'extension calc

Redéfinit les commandes `\setcounter`, `\addtocounter`, `\setlength` et `\addtolength`, qui acceptent dès lors des expressions algébriques.

Code



Fichier Beamer3/code367.tex

```
\setcounter{COMPTEUR}{3*_6/_5+_8}
\theCOMPTEUR

\setlength\LONGUEUR{1cm+_1in}
\the\LONGUEUR

\setlength\LONGUEUR{\totalheightof{I
}}
\the\LONGUEUR

\setlength\LONGUEUR{50_pt*_ (3*_ \
real{1.5})+_7*_ \ratio{13cm}{10in
}_)}
\the\LONGUEUR
```

Sortie

```
11
100.72273pt
4.82104pt
350.0pt
```

L'extension fp

Permet de faire des calculs en virgule flottante.

Limitations :

- incompatible avec `multido`;
- incompatible avec `[français]{babel}`.

Solution qui semble fonctionner : charger `fp` en premier.

L'extension fp

Permet de faire des calculs en virgule flottante.

Limitations :

- incompatible avec `multido` ;
- incompatible avec `[français]{babel}`.

Solution qui semble fonctionner : charger `fp` en premier.

Code



Fichier Beamer3/code369.tex

```
\newcommand{\x}{2}
\FPeval{\resultat}{x^2+x+1}
\resultat

\FPeval{\x}{2*pi/3}
\FPeval{\resultat}{cos(x)}
\resultat
```

Sortie

```
6.9999999999999999968
-0.50000000000000000000
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
 - Les fontes
 - Calculer en \LaTeX
 - **En finir avec les bad boxes**
 - Enlever les espaces indésirables
 - Écrire dans des fichiers
 - Divers
- 4 Programmation \LaTeX avancée

En finir avec les **bad boxes**

Algorithme de mise en page horizontale utilisé par \LaTeX

L'algorithme pour découper chaque paragraphe en lignes se fait en trois passes :

En finir avec les **bad boxes**

Algorithme de mise en page horizontale utilisé par \LaTeX

L'algorithme pour découper chaque paragraphe en lignes se fait en trois passes :

- 1 la première passe tente un ajustement simplifié (espaces) ;

En finir avec les **bad boxes**

Algorithme de mise en page horizontale utilisé par \LaTeX

L'algorithme pour découper chaque paragraphe en lignes se fait en trois passes :

- 1 la première passe tente un ajustement simplifié (espaces) ;
- 2 si la "badness" est trop importante la deuxième passe tente un ajustement plus complexe (césures) ;

En finir avec les **bad boxes**

Algorithme de mise en page horizontale utilisé par \LaTeX

L'algorithme pour découper chaque paragraphe en lignes se fait en trois passes :

- 1 la première passe tente un ajustement simplifié (espaces) ;
- 2 si la "badness" est trop importante la deuxième passe tente un ajustement plus complexe (césures) ;
- 3 si la "badness" est encore trop importante il y a une troisième passe agissant sur les césures ;

En finir avec les **bad boxes**

Algorithme de mise en page horizontale utilisé par \LaTeX

L'algorithme pour découper chaque paragraphe en lignes se fait en trois passes :

- 1 la première passe tente un ajustement simplifié (espaces) ;
- 2 si la "badness" est trop importante la deuxième passe tente un ajustement plus complexe (césures) ;
- 3 si la "badness" est encore trop importante il y a une troisième passe agissant sur les césures ;
- 4 si la "badness" est toujours trop importante c'est soit une `underfull \hbox` (ligne pas assez remplie) soit une `overfull \hbox` (ligne qui dépasse).

En finir avec les **bad boxes**

Algorithme de mise en page horizontale utilisé par \LaTeX

L'algorithme pour découper chaque paragraphe en lignes se fait en trois passes :

- 1 la première passe tente un ajustement simplifié (espaces) ;
- 2 si la "badness" est trop importante la deuxième passe tente un ajustement plus complexe (césures) ;
- 3 si la "badness" est encore trop importante il y a une troisième passe agissant sur les césures ;
- 4 si la "badness" est toujours trop importante c'est soit une `underfull \hbox` (ligne pas assez remplie) soit une `overfull \hbox` (ligne qui dépasse).

La "badness" est une quantité qui permet d'évaluer la laideur de la pire ligne d'un paragraphe :

En finir avec les **bad boxes**

Algorithme de mise en page horizontale utilisé par \LaTeX

L'algorithme pour découper chaque paragraphe en lignes se fait en trois passes :

- 1 la première passe tente un ajustement simplifié (espaces) ;
- 2 si la "badness" est trop importante la deuxième passe tente un ajustement plus complexe (césures) ;
- 3 si la "badness" est encore trop importante il y a une troisième passe agissant sur les césures ;
- 4 si la "badness" est toujours trop importante c'est soit une `underfull \hbox` (ligne pas assez remplie) soit une `overfull \hbox` (ligne qui dépasse).

La "badness" est une quantité qui permet d'évaluer la laideur de la pire ligne d'un paragraphe :

- $+\infty$ (10000) si une ligne dépasse de la page ;
- sinon c'est une expression compliquée faisant intervenir le facteur de dilatation de la colle utilisée.

En finir avec les **bad boxes**

Les règles du jeu

- Les « overfull boxes » sont potentiellement graves ;

En finir avec les **bad boxes**

Les règles du jeu

- Les « overfull boxes » sont potentiellement graves ;
- les « underfull boxes » sont laides.

En finir avec les **bad boxes**

Les règles du jeu

- Les « overfull boxes » sont potentiellement graves ;
- les « underfull boxes » sont laides.

Paramètres de design

- Taille du papier ;
- Langage ;
- Jeu de fontes ;
- Taille de fontes ;
- Indentation.

En finir avec les **bad boxes**

Les règles du jeu

- Les « overfull boxes » sont potentiellement graves ;
- les « underfull boxes » sont laides.

Paramètres de design

- Taille du papier ;
- Langage ;
- Jeu de fontes ;
- Taille de fontes ;
- Indentation.

Paramètres T_EXniques

- `\pretolerance` : borne pour la première passe ;
- `\tolerance` : borne pour la deuxième passe ;
- `\hbadness` : borne des `underfull \hbox` ;
- `\hfuzz` : longueur admissible de dépassement de la ligne ;
- `\emergencystretch` : longueur utilisée dans la troisième passe.

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

1 Initialisation :

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

1 Initialisation :

Dans le préambule

```
\emergencystretch = 0pt  
\pretolerance = 150  
\tolerance = 250  
\hbadness = 150  
\hfuzz = 0pt
```

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

1 Initialisation :

Dans le préambule

```
\emergencystretch = 0pt  
\pretolerance = 150  
\tolerance = 250  
\hbadness = 150  
\hfuzz = 0pt
```

▷ Si pas de overfull hboxes dans le log à ce stade on peut passer à l'étape 3.

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et s'il n'y a plus de overfull hboxes passer à l'étape 3, sinon :

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et s'il n'y a plus de overfull hboxes passer à l'étape 3, sinon :
 - si la plus grande overfull `\hbox` dépasse seulement d'une petite valeur (par exemple 2pt) alors `\hfuzz = 2pt` devrait régler le problème ;

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et s'il n'y a plus de overfull hboxes passer à l'étape 3, sinon :
 - si la plus grande overfull `\hbox` dépasse seulement d'une petite valeur (par exemple 2pt) alors `\hfuzz = 2pt` devrait régler le problème ;
 - sinon, il peut s'agir d'un problème de césure, qu'on peut régler avec `\-` ou `\hyphenation{hy-phe-na-tion}` ;

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- ① Initialisation.
- ② Re-compiler avec `\tolerance = 9999` et s'il n'y a plus de overfull hboxes passer à l'étape 3, sinon :
 - si la plus grande overfull `\hbox` dépasse seulement d'une petite valeur (par exemple 2pt) alors `\hfuzz = 2pt` devrait régler le problème ;
 - sinon, il peut s'agir d'un problème de césure, qu'on peut régler avec `\-` ou `\hyphenation{hy-phe-na-tion}` ;
 - si à ce stade le problème des overfull hboxes n'est pas réglé, il va falloir changer le texte ou adapter certains paramètres de mise en page (largeur,indentation. . .)

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et supprimer les overfull hboxes.
- 3 Déterminer la valeur optimale de `\tolerance` :

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et supprimer les overfull hboxes.
- 3 Déterminer la valeur optimale de `\tolerance` :
 - soit *badmax* la "badness" de la pire underfull \hbox, mettre `\tolerance = badmax` et `\hbadness = badmax - 1`;

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et supprimer les overfull hboxes.
- 3 Déterminer la valeur optimale de `\tolerance` :
 - soit *badmax* la "badness" de la pire underfull \hbox, mettre `\tolerance = badmax` et `\hbadness = badmax - 1`;
 - diminuer la valeur de `\tolerance` tant qu'il n'y a pas de overfull \hbox.

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et supprimer les overfull hboxes.
- 3 Déterminer la valeur optimale de `\tolerance`.
- 4 Si à ce stade certaines underfull hboxes sont trop laides (espaces entre les mots trop important) il reste un dernier espoir :

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et supprimer les overfull hboxes.
- 3 Déterminer la valeur optimale de `\tolerance`.
- 4 Si à ce stade certaines underfull hboxes sont trop laides (espaces entre les mots trop important) il reste un dernier espoir :
 - diminuer la valeur de `\tolerance` et mettre `\hbadness` à cette valeur moins 1;

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et supprimer les overfull hboxes.
- 3 Déterminer la valeur optimale de `\tolerance`.
- 4 Si à ce stade certaines underfull hboxes sont trop laides (espaces entre les mots trop important) il reste un dernier espoir :
 - diminuer la valeur de `\tolerance` et mettre `\hbadness` à cette valeur moins 1;
 - utiliser `\emergencystretch = 1em`;

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et supprimer les overfull hboxes.
- 3 Déterminer la valeur optimale de `\tolerance`.
- 4 Si à ce stade certaines underfull hboxes sont trop laides (espaces entre les mots trop important) il reste un dernier espoir :
 - diminuer la valeur de `\tolerance` et mettre `\hbadness` à cette valeur moins 1;
 - utiliser `\emergencystretch = 1em`;
 - si des overfull bad boxes apparaissent à ce stade il faut soit changer le document, soit augmenter `\tolerance`, soit augmenter `\emergencystretch` (mais pas trop!).

En finir avec les **bad boxes**

Algorithme à utiliser pour supprimer les bad boxes horizontales

- 1 Initialisation.
- 2 Re-compiler avec `\tolerance = 9999` et supprimer les overfull hboxes.
- 3 Déterminer la valeur optimale de `\tolerance`.
- 4 Ajuster éventuellement les paramètres pour obtenir le meilleur effet visuel.
- 5 Poser `\hbadness=10000` pour supprimer tous les messages "underfull hbox".

En finir avec les **bad boxes**

Supprimer les bad boxes verticales

Elles sont généralement plus rares, mais au cas où :

- utiliser des longueurs élastiques avec `\vspace` (par exemple `\vspace{1cm plus 2mm minus 5mm}`);

En finir avec les **bad boxes**

Supprimer les bad boxes verticales

Elles sont généralement plus rares, mais au cas où :

- utiliser des longueurs élastiques avec `\vspace` (par exemple `\vspace{1cm plus 2mm minus 5mm}`);
- `\enlargethispage{longueur}` pour augmenter (ou diminuer) la hauteur de la page courante ;

En finir avec les **bad boxes**

Supprimer les bad boxes verticales

Elles sont généralement plus rares, mais au cas où :

- utiliser des longueurs élastiques avec `\vspace` (par exemple `\vspace{1cm plus 2mm minus 5mm}`);
- `\enlargethispage{longueur}` pour augmenter (ou diminuer) la hauteur de la page courante ;
- les extensions `longtable` ou `supertabular` permettent de faire des tableaux qui peuvent être coupés entre les pages.

Empêcher les sauts dans certaines zones

- L'extension `needspace` introduit la commande `\needspace{hauteur}` qui passe à la page suivante s'il reste moins que la hauteur disponible avant la fin de la page en cours ;

Empêcher les sauts dans certaines zones

- L'extension `needspace` introduit la commande `\needspace{hauteur}` qui passe à la page suivante s'il reste moins que la hauteur disponible avant la fin de la page en cours ;
- `\interlinepenalty = 10000` au début d'une liste décourage \LaTeX d'insérer un saut de page dans les items ;
- `\@itempenalty = 10000` pour éviter les sauts de page entre les items ;

Empêcher les sauts dans certaines zones

- L'extension `needspace` introduit la commande `\needspace{hauteur}` qui passe à la page suivante s'il reste moins que la hauteur disponible avant la fin de la page en cours ;
- `\interlinepenalty = 10000` au début d'une liste décourage \LaTeX d'insérer un saut de page dans les items ;
- `\@itempenalty = 10000` pour éviter les sauts de page entre les items ;
- `\widowpenalty=1000` et `\clubpenalty=1000` pour éviter les sauts de page après la première ligne ou avant la dernière ligne d'un paragraphe.

Suggérer ou empêcher des sauts précis

On peut formuler la demande (n) avec plus au moins d'insistance : 1 pour une requête timide et 4 pour un ordre.

Par défaut c'est 4.

Suggérer ou empêcher des sauts précis

On peut formuler la demande (n) avec plus au moins d'insistance : 1 pour une requête timide et 4 pour un ordre.

Par défaut c'est 4.

- `\linebreak[n]` pour demander de couper la ligne ;

Suggérer ou empêcher des sauts précis

On peut formuler la demande (n) avec plus au moins d'insistance : 1 pour une requête timide et 4 pour un ordre.

Par défaut c'est 4.

- `\linebreak[n]` pour demander de couper la ligne ;
- `\nolinebreak[n]` pour demander de ne pas couper la ligne ;

Suggérer ou empêcher des sauts précis

On peut formuler la demande (n) avec plus au moins d'insistance : 1 pour une requête timide et 4 pour un ordre.

Par défaut c'est 4.

- `\linebreak[n]` pour demander de couper la ligne ;
- `\nolinebreak[n]` pour demander de ne pas couper la ligne ;
- `\pagebreak` et `\nopagebreak` fonctionnent de manière identique pour les sauts de page.

Suggérer ou empêcher des sauts précis

On peut formuler la demande (n) avec plus au moins d'insistance : 1 pour une requête timide et 4 pour un ordre.

Par défaut c'est 4.

- `\linebreak[n]` pour demander de couper la ligne ;
- `\nolinebreak[n]` pour demander de ne pas couper la ligne ;
- `\pagebreak` et `\nopagebreak` fonctionnent de manière identique pour les sauts de page.

Code



Fichier Beamer3/code389.tex

```
\large
Du texte à couper lorsque n = \
  \linebreak[1]1

Du texte à couper lorsque n = \
  \linebreak[2]2

Du texte à couper lorsque n = \
  \linebreak[3]3

Du texte à couper lorsque n = \
  \linebreak[4]4
```

Sortie

```
Du texte à couper lorsque n = 1
Du texte à couper lorsque n =
2
Du texte à couper lorsque n =
3
Du texte à couper lorsque n =
4
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
 - Les fontes
 - Calculer en \LaTeX
 - En finir avec les **bad boxes**
 - **Enlever les espaces indésirables**
 - Écrire dans des fichiers
 - Divers
- 4 Programmation \LaTeX avancée

Commentaires de fin de ligne

Dans les définitions de commandes, ils permettent à la fois d'indenter proprement le code et d'éviter les espaces ajoutés par les sauts de ligne.

Commentaires de fin de ligne

Dans les définitions de commandes, ils permettent à la fois d'indenter proprement le code et d'éviter les espaces ajoutés par les sauts de ligne.

Code



Fichier Beamer3/code391.tex

```
\newcommand\PASBIEN [1]{  
  \textbf{#1}  
}  
\newcommand\BIEN [1]{%  
  \textbf{#1}%  
}
```

```
Sans \commentaire (\PASBIEN{argument})  
  \avec \commentaires (\BIEN{  
    argument}).
```

Sortie

Sans commentaire (**argument**) avec
commentaires (**argument**).

Ignorer les espaces qui suivent

L'utilisateur des commandes n'a pas forcément envie de taper des commentaires à la fin de chaque ligne.

`\ignorespaces` permet d'ignorer les espaces qui suivent un appel de commande (ou le début d'un environnement).

Ignorer les espaces qui suivent

L'utilisateur des commandes n'a pas forcément envie de taper des commentaires à la fin de chaque ligne.

`\ignorespaces` permet d'ignorer les espaces qui suivent un appel de commande (ou le début d'un environnement).

Code



Fichier Beamer3/code393.tex

```
\newenvironment{EVIDENCE1}{\bfseries  
  }{  
}  
\newenvironment{EVIDENCE2}{\bfseries\  
  ignorespaces}{  
}  
  
\begin{itemize}  
  \item un mot \begin{EVIDENCE1}  
    décallé;  
  \end{EVIDENCE1}  
  \item un mot \begin{EVIDENCE2}  
    pas décallé;  
  \end{EVIDENCE2}  
\end{itemize}
```

Sortie

- un mot **décallé**;
- un mot **pas décallé**;

Ignorer les espaces après un environnement

`\ignorespacesafterend` permet d'ignorer les espaces qui suivent après l'utilisation de `\end{...}`.

Ignorer les espaces après un environnement

`\ignorespacesafterend` permet d'ignorer les espaces qui suivent après l'utilisation de `\end{...}`.

Code



Fichier Beamer3/code395.tex

```
\newenvironment{EVIDENCE3}{\bfseries\ignorespaces}{}  
\newenvironment{EVIDENCE4}{\bfseries\ignorespaces}{\ignorespacesafterend}  
  
\begin{itemize}  
  \item\unpoint\begin{EVIDENCE3}  
    décallé  
  \end{EVIDENCE3}  
;  
  \item\unpoint\begin{EVIDENCE4}  
    pas décallé  
  \end{EVIDENCE4}  
;  
\end{itemize}
```

Sortie

- un point **décallé** ;
- un point **pas décallé** ;

Enlever les espaces qui ont déjà été lus

`\unskip` (en mode horizontal) permet d'enlever les espaces qui viennent d'être tapés.

Enlever les espaces qui ont déjà été lus

`\unskip` (en mode horizontal) permet d'enlever les espaces qui viennent d'être tapés.

Code



Fichier Beamer3/code397.tex

```
\newenvironment{EVIDENCE5}{\bfseries\ignorespaces}{.}
\newenvironment{EVIDENCE6}{\bfseries\ignorespaces}{\unskip.}

\begin{itemize}
  \item un point \begin{EVIDENCE5}
    décallé
  \end{EVIDENCE5}
  \item un point \begin{EVIDENCE6}
    pas décallé
  \end{EVIDENCE6}
\end{itemize}
```

Sortie

- un point **décallé** .
- un point **pas décallé**.

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
 - Les fontes
 - Calculer en \LaTeX
 - En finir avec les **bad boxes**
 - Enlever les espaces indésirables
 - **Écrire dans des fichiers**
 - Divers
- 4 Programmation \LaTeX avancée

L'extension filecontents

Elle définit l'environnement `\begin{filecontents}{fichier}` qui écrit son contenu tel quel dans le fichier.

La variante étoilée `filecontents*` n'ajoute pas de commentaires au début.

L'extension filecontents

Elle définit l'environnement `\begin{filecontents}{fichier}` qui écrit son contenu tel quel dans le fichier.

La variante étoilée `filecontents*` n'ajoute pas de commentaires au début.

Code



Fichier Beamer3/code399.tex

```
\begin{filecontents}{test.tex}
Voici le contenu du fichier.
Voici le contenu du fichier.
Voici le contenu du fichier.
\end{filecontents}
\verbatiminput{test.tex}
```

Sortie

```
%% LaTeX2e file 'test.tex'
%% generated by the 'filecontents' environm
%% from source 'Beamer3' on 2010/03/18.
%%
Voici le contenu du fichier.
Voici le contenu du fichier.
Voici le contenu du fichier.
```

Les registres d'écriture

- `\newwrite\registre` définit un `\registre` d'écriture;
- `\openout\registre fichier` ouvre un fichier en écriture;
- `\write\registre{...}` écrit une ligne dans le fichier ouvert;
- `\closeout\registre` ferme le fichier ouvert après usage.

Les registres d'écriture

- `\newwrite\registre` définit un `\registre` d'écriture;
- `\openout\registre` fichier ouvre un fichier en écriture;
- `\write\registre{...}` écrit une ligne dans le fichier ouvert;
- `\closeout\registre` ferme le fichier ouvert après usage.

Précautions d'utilisation :

- toujours refermer un fichier après usage;
- faire précéder `\openout`, `\write` et `\closeout` de `\immediate`;
- gare aux caractères spéciaux!

Les registres d'écriture

Exemple d'utilisation

Code



Fichier Beamer3/code400.tex

```
\newwrite\MONWRITE
\immediate\openout\MONWRITE\test.tex
\immediate\write\MONWRITE{Salut ,}
\immediate\write\MONWRITE{}
\immediate\write\MONWRITE{Je suis un
  fichier généré par \noexpand\
  LaTeX}
\immediate\write\MONWRITE{Les \
  noexpand\commandes non protégées}
\immediate\write\MONWRITE{son
  développées:\the\textwidth.}
\immediate\write\MONWRITE{Caractères
  spéciaux:\&,\#,\{,\}}
\immediate\closeout\MONWRITE
\verbatiminput{test.tex}
```

Sortie

Salut,

Je suis un fichier généré par \LaTeX
Les \commandes non protégées
sont développées: 142.88866pt.
Caractères spéciaux: &, ##, {, }

Les registres d'écriture

Registres prédéfinis

- `\write-1{...}` pour écrire dans le log ;
- `\write17{...}` pour écrire dans le log et dans le terminal ;
- `\write18{...}` pour transmettre une commande au système d'exploitation, comme si elle était tapée directement dans le terminal.

Les registres d'écriture

Registres prédéfinis

- `\write-1{...}` pour écrire dans le log ;
- `\write17{...}` pour écrire dans le log et dans le terminal ;
- `\write18{...}` pour transmettre une commande au système d'exploitation, comme si elle était tapée directement dans le terminal.

La commande `\write18` étant **dangereuse**, elle est désactivée par défaut. On peut l'activer avec `-shell-escape` sous Linux ou `--enable-write18` sous Windows/MikTeX.

Les registres d'écriture

Registres prédéfinis

- `\write-1{...}` pour écrire dans le log ;
- `\write17{...}` pour écrire dans le log et dans le terminal ;
- `\write18{...}` pour transmettre une commande au système d'exploitation, comme si elle était tapée directement dans le terminal.

La commande `\write18` étant **dangereuse**, elle est désactivée par défaut. On peut l'activer avec `-shell-escape` sous Linux ou `--enable-write18` sous Windows/MikTeX.

Code



Fichier Beamer3/code403.tex

```
\immediate\write18{ls_*r.*>_test.txt  
}  
\verbatiminput{test.txt}
```

Sortie

```
Coeur.txt  
beamer.tex  
seminar.aux  
seminar.con  
seminar.ps  
seminar.tex  
seminar.tmp
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
 - Les fontes
 - Calculer en \LaTeX
 - En finir avec les **bad boxes**
 - Enlever les espaces indésirables
 - Écrire dans des fichiers
 - Divers
- 4 Programmation \LaTeX avancée

Message d'erreur : no room for a new `\dimen/\count`

Certaines classes ou extensions récentes utilisent beaucoup de registres de compteurs et dimensions, or \LaTeX n'en a que 256.

Message d'erreur : `no room for a new \dimen/\count`

Certaines classes ou extensions récentes utilisent beaucoup de registres de compteurs et dimensions, or \LaTeX n'en a que 256.

Solution : l'extension `etex` augmente le nombre de registres disponibles.

La lettre @

Certaines commandes \LaTeX de bas niveau sont protégées : leur nom contient le caractère @.

Par exemple `\@title`, `\@author` et `\@date` contiennent habituellement les informations de la page de titre du document.

La lettre @

Certaines commandes \LaTeX de bas niveau sont protégées : leur nom contient le caractère @.

Par exemple `\@title`, `\@author` et `\@date` contiennent habituellement les informations de la page de titre du document.

- `\makeatletter` rend @ semblable aux autres lettres, et permet d'accéder aux commandes protégées ;
- `\makeatother` restaure le statut habituel de @.

La lettre @

Certaines commandes \LaTeX de bas niveau sont protégées : leur nom contient le caractère @.

Par exemple `\@title`, `\@author` et `\@date` contiennent habituellement les informations de la page de titre du document.

- `\makeatletter` rend @ semblable aux autres lettres, et permet d'accéder aux commandes protégées ;
- `\makeatother` restaure le statut habituel de @.

Code



Fichier Beamer3/code406.tex

```
\makeatletter
Ce document a été écrit le \@date.
\makeatother
```

Sortie

Ce document a été écrit le 18 mars 2010.

Longueurs rapides

`\longueur=...` permet de stocker une valeur dans une longueur, pendant la durée du groupe en cours.

Longueurs rapides

`\longueur=...` permet de stocker une valeur dans une longueur, pendant la durée du groupe en cours.

Attention au texte qui suit : il est préférable de mettre un `\relax` après la valeur à stocker.

Longueurs rapides

`\longueur=...` permet de stocker une valeur dans une longueur, pendant la durée du groupe en cours.

Attention au texte qui suit : il est préférable de mettre un `\relax` après la valeur à stocker.

Code



Fichier Beamer3/code409.tex

```
\begin{itemize}
  \itemsep=1cm\relax
  \item Un;
  \item deux;
  \item trois.
\end{itemize}
```

Sortie

- Un;

- deux;

- trois.

Afficher une longueur

`\the\longueur` affiche la valeur (en points) d'une longueur.

Code



Fichier Beamer3/code410.tex

```
Largeur du texte : \the\textwidth.
```

Sortie

Largeur du texte : 142.88866pt.

Correction italique

Dans la mesure du possible, utiliser `\textit` au lieu de `\itshape`.

Sinon, il peut être nécessaire d'ajouter manuellement une correction italique `\/` avant le passage au mode normal.

Correction italique

Dans la mesure du possible, utiliser `\textit` au lieu de `\itshape`.

Sinon, il peut être nécessaire d'ajouter manuellement une correction italique `\` avant le passage au mode normal.

Code



Fichier Beamer3/code412.tex

```
\Huge  
\textit{I}talique  
  
\itshape I\upshape talique  
  
\itshape I\\ \upshape talique
```

Sortie

*I*talique
*I*talique
*I*talique

Ajouter du texte à une commande

`\addto\commande{...}` ajoute du contenu à la fin de la `\commande`.

Par exemple, `\addto\maketitle{...}` permet de rajouter une ligne dans la page de titre.

Ajouter du texte à une commande

`\addto\commande{...}` ajoute du contenu à la fin de la `\commande`.

Par exemple, `\addto\maketitle{...}` permet de rajouter une ligne dans la page de titre.

Précautions d'emploi :

- si la `\commande` n'existe pas elle est définie comme vide juste avant l'ajout ;
- si elle existe déjà, elle ne doit pas avoir de paramètres.

Ajouter du texte à une commande

`\addto\commande{...}` ajoute du contenu à la fin de la `\commande`.

Par exemple, `\addto\maketitle{...}` permet de rajouter une ligne dans la page de titre.

Précautions d'emploi :

- si la `\commande` n'existe pas elle est définie comme vide juste avant l'ajout ;
- si elle existe déjà, elle ne doit pas avoir de paramètres.

Code



Fichier Beamer3/code417.tex

```
\addto\bla{bla}  
\addto\bla{bla}  
\addto\bla{bla}  
  
\bla
```

Sortie

blablabla

Traduction des titres avec babel

À chaque changement de langue, babel utilise `\captions...` pour changer les noms de titres.

Il suffit d'ajouter à la commande correspondante les titres désirés.

Traduction des titres avec babel

À chaque changement de langue, babel utilise `\captions...` pour changer les noms de titres.

Il suffit d'ajouter à la commande correspondante les titres désirés.

Exemples



Fichier Beamer3/code420.tex

```
%Pour le français
\addto\captionsfrench{%
\renewcommand{\refname}{Bibliographie}%
\renewcommand{\contentsname}{Sommaire}%
}
```

```
%Pour l'allemand
\addto\captionsgerman{%
\renewcommand{\refname}{...}%
\renewcommand{\contentsname}{...}%
}
```

Exercice

Pourquoi le code suivant ne produit-il pas d'erreur, bien qu'on n'ait pas utilisé `\makeatletter` ?

Code



Fichier Beamer3/code421.tex

```
Ce_document_a_été_écrit_le_\@date.
```

Solution

Pourquoi le code suivant ne produit-il pas d'erreur, bien qu'on n'ait pas utilisé `\makeatletter` ?

La commande `\@` existe déjà (elle a pour effet d'agir sur la taille des espaces), donc le code est compris comme :

Sortie

Ce document a été écrit le date.

Exercice

Donner un code à mettre dans le préambule qui permette d'ajouter l'adresse email de l'auteur avec `\email{...}` dans la page de titre d'un article

Solution

Donner un code à mettre dans le préambule qui permette d'ajouter l'adresse email de l'auteur avec `\email{...}` dans la page de titre d'un article

Code



Fichier Beamer3/code422.tex

```
\usepackage{url}

\makeatletter

\newcommand\@email{}
\newcommand\email[1]{%
  \u\renewcommand\@email{\url{#1}}%
}

\addto\maketitle{%
  \u\begin{center}%
  \u\@email
  \u\end{center}%
}

\makeatother
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée
 - Tests et branchements conditionnels
 - Écrire une nouvelle extension
 - Écrire une nouvelle classe
 - Définition avancée des commandes
 - Boucles
 - Codes de catégorie

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée
 - Tests et branchements conditionnels
 - Écrire une nouvelle extension
 - Écrire une nouvelle classe
 - Définition avancée des commandes
 - Boucles
 - Codes de catégorie

Les \if...

On peut créer un nouveau \if avec \newif\ifXXX, qui va définir trois commandes :

- \ifXXX, qu'on doit faire suivre de \fi (éventuellement après une clause \else);

Les \if...

On peut créer un nouveau \if avec \newif\ifXXX, qui va définir trois commandes :

- \ifXXX, qu'on doit faire suivre de \fi (éventuellement après une clause \else);
- \XXXtrue pour rendre le test vrai dans le groupe en cours;
- \XXXfalse pour le rendre faux (c'est la valeur par défaut après \newif).

Les \if...

On peut créer un nouveau \if avec \newif\ifXXX, qui va définir trois commandes :

- \ifXXX, qu'on doit faire suivre de \fi (éventuellement après une clause \else);
- \XXXtrue pour rendre le test vrai dans le groupe en cours;
- \XXXfalse pour le rendre faux (c'est la valeur par défaut après \newif).

Code



Fichier Beamer3/code425.tex

```
\newif\ifmachin
\newcommand\machin[2]{%
  uu\ifmachin
  uuuu#1\\%
  uu\else
  uuuu#2\\%
  uu\fi
}%
\machin{Vrai}{Faux}
{\machintrue
\machin{Vrai}{Faux}}
\machin{Vrai}{Faux}
```

Sortie

```
Faux
Vrai
Faux
```

Les \if...

Tests sur les valeurs

L^AT_EX contient un certain nombre de \if... prédéfinis pour tester des valeurs numériques :

- \ifodd nombre pour tester la parité :

Code



Fichier Beamer3/code426.tex

```
\newcounter{XXX}

%Deviner ce qui se passe sans le \relax
\newcommand\parity[1]{%
  \ifodd#1\relax
  \#1 est impair.%
  \else
  \#1 est pair.%
  \fi
}

\parity{10}
\parity{\theXXX}
\setcounter{XXX}{5}\parity{\arabic{XXX}}
```

Sortie

10 est pair. 0 est pair. 5 est impair.

Les \if...

Tests sur les valeurs

L^AT_EX contient un certain nombre de \if... prédéfinis pour tester des valeurs numériques :

- \ifodd nombre pour tester la parité;
- \ifnum num1 op num2 pour comparer deux valeurs (op peut être =, < ou >) :

Code



Fichier Beamer3/code434.tex

```
\newcommand\compare[2]{%
  \ifnum#1=#2\relax#1est égal à
    #2\else
  \ifnum#1>#2\relax
    #1est plus grand que #2%
  \else
    #1est plus petit que #2%
  \fi
}\fi.
}

\compare{-50}{-50}
\compare{3}{+4}
\compare{\theXXX}{100}
```

Sortie

-50 est égal à -50. 3 est plus petit que +4. 5 est plus petit que 100.

Les \if...

Tests sur les valeurs

L^AT_EX contient un certain nombre de \if... prédéfinis pour tester des valeurs numériques :

- \ifodd nombre pour tester la parité;
- \ifnum num1 op num2 pour comparer deux valeurs;
- \ifdim dim1 # dim2 pour comparer deux longueurs;

Code



Fichier Beamer3/code441.tex

```
\newcommand\PROPORTIONS [1]{%  
  \settowidth\WIDTH{#1}%  
  \settowidth\HEIGHT{#1}%  
  \ifdim\WIDTH>\HEIGHT\relax  
    #1" est plus large que haut%  
  \else  
    #1" est plus haut que large%  
  \fi  
}  
  
\PROPORTIONS{I}, \PROPORTIONS{w}.
```

Sortie

"I" est plus haut que large, "w" est plus large que haut.

Les \if...

Tests sur les valeurs

L^AT_EX contient un certain nombre de \if... prédéfinis pour tester des valeurs numériques :

- \ifodd nombre pour tester la parité;
- \ifnum num1 op num2 pour comparer deux valeurs;
- \ifdim dim1 # dim2 pour comparer deux longueurs;
- \ifcase num ... \or ... \or ... \else ... \fi pour effectuer une action différente si num vaut 0, 1, 2... ou une autre valeur :

Code



Fichier Beamer3/code449.tex

```
\newcommand\numdujour [1]{%  
  \ifcase#1\relax\ERREUR\or_lundi\or_  
    mardi\or_mercredi\or_jeudi\or_  
    vendredi\or_samedi\or_dimanche\  
    else\ERREUR\fi  
}  
  
\numdujour{0},\numdujour{1},\  
  numdujour{7},\numdujour{-3}.
```

Sortie

ERREUR, lundi, dimanche, ERREUR.

Les \if...

Tests sur le mode en cours

- \ifhmode pour tester si on est en mode horizontal :

Code



Fichier Beamer3/code450.tex

```
\newcommand\BIDULE[1]{%  
  \ifitshape#1%  
  \ifhmode  
    \unskip  
    \/  
  \fi}%  
\Large  
  
\BIDULE{ }fonctionne.  
  
\BIDULE{FONCTIONNE}plutôt bien.  
  
\ifitshape FONCTIONNE pas.
```

Sortie

fonctionne.

FONCTIONNE plutôt bien.

FONCTIONNE pas.

Les \if...

Tests sur le mode en cours

- \ifhmode pour tester si on est en mode horizontal ;
- \ifvmode pour tester si on est en mode vertical :

Code



Fichier Beamer3/code454.tex

```
\newcommand\SHOWVMODE{\ifvmode[
  vertical]\else[horizontal]\fi}

\SHOWVMODE\en_début_de_paragraphe,\
SHOWVMODE\au_milieu.

\begin{itemize}
\item Dans une liste :
\item \SHOWVMODE\au_début_d'un_item
,\SHOWVMODE\au_milieu.
\end{itemize}\SHOWVMODE\à_la_fin_de
la_liste.
```

Sortie

[vertical] en début de paragraphe, [horizontal] au milieu.

- Dans une liste :
- [vertical] au début d'un item,
[horizontal] au milieu.

[vertical] à la fin de la liste.

Les \if...

Tests sur le mode en cours

- \ifhmode pour tester si on est en mode horizontal ;
- \ifvmode pour tester si on est en mode vertical ;
- \ifmmode pour tester si on est en mode mathématique :

Code



Fichier Beamer3/code458.tex

```
\newcommand\DOLLAR{%  
  \ifmmode\mathdollar\else\textdollar  
  \fi  
}  
  
\DOLLAR\_(en_mode_ttexte),_\$\DOLLAR+1$\br/>  _(en_mode_math)_et  
  \[\sum_{\DOLLAR}\]  
(en_mode_display).
```

Sortie

\$ (en mode texte), \$ + 1 (en mode math)
et

$$\sum_s$$

(en mode display).

Les `\if...`

Tests sur le mode en cours

- `\ifhmode` pour tester si on est en mode horizontal ;
- `\ifvmode` pour tester si on est en mode vertical ;
- `\ifmmode` pour tester si on est en mode mathématique ;
- `\ifinner` pour tester si on est en mode encapsulé (mathématiques en mode texte ou texte sans paragraphe associé comme `\hbox` ou `\vbox`).

Les \if...

Tests supplémentaires

- \iftrue est toujours vrai :

Code



Fichier Beamer3/code462.tex

```
\iftrue  
  C'est vrai  
\else  
  C'est faux  
\fi
```

Sortie

C'est vrai

Les \if...

Tests supplémentaires

- \iftrue est toujours vrai;
- \iffalse est toujours faux :

Code



Fichier Beamer3/code466.tex

```
\iffalse
  C'est vrai
\else
  C'est faux
\fi
```

Sortie

C'est faux

Les \if...

Tests supplémentaires

- \iftrue est toujours vrai;
- \iffalse est toujours faux;
- \IfFileExists{fichier}{action1}{action2} effectue action1 ou action2 selon que le fichier existe ou non :

Code



Fichier Beamer3/code470.tex

```
\IfFileExists{\jobname.bidon}{\
  jobname.bidon_existe}{\jobname.
  bidon_n'existe_pas},
\IfFileExists{\jobname.tex}{\jobname.
  tex_existe}{\jobname.tex_n'existe
  _pas}.
```

Sortie

Beamer3.bidon n'existe pas, Beamer3.tex existe.

Les \if...

Tests sur les commandes

- `\ifcsname nom\endcsname` pour tester si la commande `\nom` existe :

Code



Fichier Beamer3/code471.tex

```
\ifcsname␣only\endcsname  
␣␣Ce␣document␣utilise␣beamer.  
\else  
␣␣Ce␣document␣n'utilise␣pas␣beamer.  
\fi
```

Sortie

Ce document utilise beamer.

Les \if...

Tests sur les commandes

- `\ifcsname nom\endcsname` pour tester si la commande `\nom` existe ;
- `\ifdefined\commande` pour tester si la commande `\commande` existe :

Code



Fichier Beamer3/code475.tex

```
\ifdefined\letter\relax
\documentestunelettre.
\else
\documentn'estpasunelettre.
\fi
```

Sortie

Ce document n'est pas une lettre.

Les \if...

Tests sur les commandes

- `\ifcsname nom\endcsname` pour tester si la commande `\nom` existe ;
- `\ifdefined\commande` pour tester si la commande `\commande` existe ;
- `\ifx\commande1\commande2` pour tester si `\commande1` et `\commande2` sont **exactement** identiques :

Code



Fichier Beamer3/code479.tex

```
\newcommand\AAA{Bonjour}
\newcommand\BBB{Bonjour}
\newcommand\CCC{Bonjour!}
\newcommand\DDD{{Bonjour}}
\let\EEE\AAA
\newcommand\COMPARE[2]{\ifx#1#2\relax
  \égal\else\pas\égal\fi}

\COMPARE{\AAA}{\BBB},\COMPARE{\AAA
}{\BBB},\COMPARE{\AAA}{\CCC},\
\COMPARE{\AAA}{\DDD},\COMPARE{\
AAA}{\EEE}.
```

Sortie

égal, égal, pas égal, pas égal, égal.

Les \if...

Tests supplémentaires introduits par des extensions

- L'extension `ifpdf` introduit `\ifpdf` pour tester si on est en train de compiler avec \LaTeX ou Pdf \LaTeX :

Code



Fichier Beamer3/code480.tex

```
\ifpdf
  Ce document a été compilé en
  utilisant Pdf $\LaTeX$ .
\else
  Ce document a été compilé en
  utilisant  $\LaTeX$ .
\fi
```

Sortie

Ce document a été compilé en utilisant \LaTeX .

Les \if...

Tests supplémentaires introduits par des extensions

- L'extension `ifpdf` introduit `\ifpdf` pour tester si on est en train de compiler avec \LaTeX ou Pdf\LaTeX ;
- l'extension `changepage` (récente) introduit `\ifoddpage` pour tester si la page en cours est impaire (à faire précéder de `\checkoddpage` avant chaque utilisation).

En effet, dans certains cas, `\ifodd\value{page}\relax` ne fonctionne pas.

L'option `[strict]` de `changepage` permet de régler le problème.

Code



Fichier Beamer3/code483.tex

```
\newcommand\AFFICHE{\checkoddpage\  
  ifoddpage␣Impaire\else␣Paire\fi}  
  
\AFFICHE.\pause
```

Sortie

Impaire.

Les \if...

Tests supplémentaires introduits par des extensions

- L'extension `ifpdf` introduit `\ifpdf` pour tester si on est en train de compiler avec \LaTeX ou Pdf\LaTeX ;
- l'extension `changepage` (récente) introduit `\ifoddpage` pour tester si la page en cours est impaire (à faire précéder de `\checkoddpage` avant chaque utilisation).

En effet, dans certains cas, `\ifodd\value{page}\relax` ne fonctionne pas.

L'option `[strict]` de `changepage` permet de régler le problème.

Code



Fichier Beamer3/code485.tex

```
\newcommand\AFFICHE{\checkoddpage\  
  ifoddpage␣Impaire\else␣Paire\fi}  
  
\AFFICHE.\pause
```

Sortie

Paire.

Les `\if...`

Tests supplémentaires introduits par des extensions

- L'extension `ifpdf` introduit `\ifpdf` pour tester si on est en train de compiler avec \LaTeX ou $\text{Pdf}\LaTeX$;
- l'extension `changepage` (récente) introduit `\ifoddpage` pour tester si la page en cours est impaire (à faire précéder de `\checkoddpage` avant chaque utilisation);
- l'extension `chnpage` (ancienne) fait la même chose avec `\ifcpoddpage`.

Les \if...

L'extension ifthenelse

- `\ifthenelse{condition}{vrai}{faux}` :

Code



Fichier Beamer3/code488.tex

```
\ifthenelse{1<2}{Vrai}{Faux}
```

```
\ifthenelse{1>2}{Vrai}{Faux}
```

Sortie

Vrai

Faux

Les \if...

L'extension ifthenelse

- `\ifthenelse{condition}{vrai}{faux};`
- `\boolean{XXX}` permet de tester `\ifXXX` :

Code



Fichier Beamer3/code497.tex

```
\ifthenelse{\boolean{true}}{Vrai}{Faux}  
  
\ifthenelse{\boolean{false}}{Vrai}{Faux}  
  
\ifthenelse{\boolean{pdf}}{Vrai}{Faux  
}
```

Sortie

```
Vrai  
Faux  
Faux
```

Les \if...

L'extension ifthenelse

- `\ifthenelse{condition}{vrai}{faux};`
- `\boolean{XXX}` permet de tester `\ifXXX`;
- `\isodd{nombre}` pour tester si nombre est pair :

Code



Fichier Beamer3/code506.tex

```
\ifthenelse{\isodd{1}}{Impair}{Pair}  
  
\ifthenelse{\isodd{2}}{Impair}{Pair}
```

Sortie

```
Impair  
Pair
```

Les \if...

L'extension ifthenelse

- `\ifthenelse{condition}{vrai}{faux};`
- `\boolean{XXX}` permet de tester `\ifXXX`;
- `\isodd{nombre}` pour tester si nombre est pair;
- `\lengthtest{longueur1 op longueur2}` pour comparer des longueurs :

Code



Fichier Beamer3/code515.tex

```
\ifthenelse{\lengthtest{1pt<2pt}}{  
  Plus_petit}{Plus_grand}  
  
\ifthenelse{\lengthtest{2pt<1pt}}{  
  Plus_petit}{Plus_grand}
```

Sortie

```
Plus petit  
Plus grand
```

Les \if...

L'extension ifthenelse

- `\ifthenelse{condition}{vrai}{faux};`
- `\boolean{XXX}` permet de tester `\ifXXX`;
- `\isodd{nombre}` pour tester si nombre est pair;
- `\lengthtest[longueur1 op longueur2]` pour comparer des longueurs;
- `\isundefined{\commande}` pour tester si `\commande` est indéfinie :

Code



Fichier Beamer3/code524.tex

```
\ifthenelse{\isundefined{\BIDON}}{  
  Indéfinie}{Définie}  
  
\ifthenelse{\isundefined{\verb}}{  
  Indéfinie}{Définie}
```

Sortie

Indéfinie
Définie

Les \if...

L'extension ifthenelse

- `\ifthenelse{condition}{vrai}{faux};`
- `\boolean{XXX}` permet de tester `\ifXXX`;
- `\isodd{nombre}` pour tester si nombre est pair;
- `\lengthtest[longueur1 op longueur2]` pour comparer des longueurs;
- `\isundefined{\commande}` pour tester si `\commande` est indéfinie;
- `\equal{texte1}{texte2}` pour tester si `texte1` est égal à `texte2` (après expansion) :

Code



Fichier Beamer3/code533.tex

```
\newcommand\XXX{Bonjour}  
\newcommand\YYY{Bon}  
\newcommand\ZZZ{jour}  
  
\ifthenelse{\equal{{bonjour}}{bonjour}}{\'  
Egal}{Différent}  
  
\ifthenelse{\equal{\XXX}{\YYY\ZZZ}}{\'  
Egal}{Différent}
```

Sortie

Différent

Égal

Les \if...

L'extension ifthenelse

- `\ifthenelse{condition}{vrai}{faux};`
- `\boolean{XXX}` permet de tester `\ifXXX`;
- `\isodd{nombre}` pour tester si nombre est pair;
- `\lengthtest[longueur1 op longueur2]` pour comparer des longueurs;
- `\isundefined{\commande}` pour tester si `\commande` est indéfinie;
- `\equal{texte1}{texte2}` pour tester si `texte1` est égal à `texte2` (après expansion);
- `\not`, `\or`, `\and` (ou `\NOT`, `\OR`, `\AND`) pour faire des opérations booléennes :

Code



Fichier Beamer3/code542.tex

```
\ifthenelse{1>3\or\boolean{false}\or\not_3<1}{Vrai}{Faux}

\ifthenelse{1<3\and\boolean{true}\and\not_1<3}{Vrai}{Faux}
```

Sortie

Vrai
Faux

Les \if...

L'extension ifthenelse

- `\ifthenelse{condition}{vrai}{faux}`;
- `\boolean{XXX}` permet de tester `\ifXXX`;
- `\isodd{nombre}` pour tester si nombre est pair;
- `\lengthtest[longueur1 op longueur2]` pour comparer des longueurs;
- `\isundefined{\commande}` pour tester si `\commande` est indéfinie;
- `\equal{texte1}{texte2}` pour tester si `texte1` est égal à `texte2` (après expansion);
- `\not`, `\or`, `\and` (ou `\NOT`, `\OR`, `\AND`) pour faire des opérations booléennes;
- `\(` et `\)` pour grouper des expressions dans des calculs :

Code



Fichier Beamer3/code551.tex

```
\ifthenelse{\not_1<2_\or_1<2}{Vrai}{Faux}
```

```
\ifthenelse{\not_(1<2_\or_1<2\)}{Vrai}{Faux}
```

Sortie

Vrai
Faux

Exercice

Définir un environnement `ignore` qui permette d'ignorer tout ce qui se trouve entre son début et sa fin.

Solution

Définir un environnement ignore qui permette d'ignorer tout ce qui se trouve entre son début et sa fin.

Code



Fichier Beamer3/code562.tex

```
%Remarque: l'environnement "comment" de l'extension verbatim est
beaucoup plus robuste!
\newif\ifignore
\ignoretrue

\newenvironment{ignore}{%
  \ifignore
\else
}{%
\fi
}
```

Exercice

Définir une commande `\cardinal{nombre}` qui affiche le nombre en respectant les conventions typographiques françaises : . . . , -2, -1, zéro, un, deux, . . . , huit, neuf, 10, 11, . . . , 98, 99, cent, 101, . . . , 999, mille, 1001, . . .

Solution

Définir une commande `\cardinal{nombre}` qui affiche le nombre en respectant les conventions typographiques françaises : ... , -2, -1, zéro, un, deux, ... , huit, neuf, 10, 11, ... , 98, 99, cent, 101, ... , 999, mille, 1001, ...

Code



Fichier Beamer3/code553.tex

```
\newcommand\cardinal[1]{%
  \ifcase#1\relax\zéro\or\un\or\deux\or\trois\or\quatre\or\cinq\or
    six\or\sept\or\huit\or\neuf\else
  \ifnum#1=100\relax
  \cent%
  \else
  \ifnum#1=1000\relax
  \mille%
  \else
  \ifnum#1%
  \fi
  \fi
  \fi
}
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée
 - Tests et branchements conditionnels
 - **Écrire une nouvelle extension**
 - Écrire une nouvelle classe
 - Définition avancée des commandes
 - Boucles
 - Codes de catégorie

Exemple minimaliste

Fichier monextension.sty



Fichier Beamer3/code554.tex

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{monextension}[2009/04/17 Ma première extension, V
1.0]

%Extensions
\RequirePackage{graphicx,url,fancybox}

%Commandes de l'extension
\newcommand\encadre[1]{\ovalbox{#1}}

\DeclareFixedFont\gotnormal{U}{yfrak}{m}{n}{10pt}
\DeclareTextFontCommand\textgot{\gotnormal}
```

Les bases

Le nom d'un fichier d'extension se termine par l'extension `.sty` et doit être accessible à \LaTeX .

Le nom d'un fichier d'extension se termine par l'extension `.sty` et doit être accessible à \LaTeX .

Rien de particulier, à part :

- le fichier doit commencer par

Code



Fichier Beamer3/code556.tex

```
\NeedsTeXFormat{LaTeX2e}  
\ProvidesPackage{nom}[Date>Description_courte,Version]
```

Le nom d'un fichier d'extension se termine par l'extension `.sty` et doit être accessible à \LaTeX .

Rien de particulier, à part :

- le fichier doit commencer par

Code



Fichier Beamer3/code557.tex

```
\NeedsTeXFormat{LaTeX2e}  
\ProvidesPackage{nom}[Date>Description_courte,Version]
```

- le caractère `@` n'est pas protégé ;

Le nom d'un fichier d'extension se termine par l'extension `.sty` et doit être accessible à \LaTeX .

Rien de particulier, à part :

- le fichier doit commencer par

Code



Fichier Beamer3/code558.tex

```
\NeedsTeXFormat{LaTeX2e}  
\ProvidesPackage{nom}[Date>Description_courte,Version]
```

- le caractère `@` n'est pas protégé ;
- pour charger une autre extension on utilise non pas `\usepackage` mais `\RequirePackage[options]{extension}`.

Les options basiques

- `\DeclareOption{nom}{action}` déclare une option de l'extension, qui effectue une action donnée.

Exemple



Fichier Beamer3/code559.tex

```
\newif\ifpretty  
\DeclareOption{pretty}{\prettytrue}
```

Les options basiques

- `\DeclareOption{nom}{action}` déclare une option de l'extension, qui effectue une action donnée.

Exemple



Fichier Beamer3/code561.tex

```
\newif\ifpretty
\DeclareOption{pretty}{\prettytrue}
```

- `\ProcessOptions` (à faire suivre de `\relax`) lance effectivement les actions prévues en fonction des options de chargement de l'extension définies par l'utilisateur.

Exemple



Fichier Beamer3/code562.tex

```
%Déclaration des actions :
\DeclareOption...
\DeclareOption...

%Exécution des actions :
\ProcessOptions\relax
```

Les options avancées

- `\DeclareOption*{action}` déclare une action par défaut pour toutes les autres options non déclarées (on peut récupérer le nom de l'option en cours avec `\CurrentOption`);
- `\PassOptionsToPackage{options}{extension}` permet d'ajouter des options supplémentaires lorsqu'on chargera `extension` avec `\RequirePackage`.

Les options avancées

- `\DeclareOption*{action}` déclare une action par défaut pour toutes les autres options non déclarées (on peut récupérer le nom de l'option en cours avec `\CurrentOption`);
- `\PassOptionsToPackage{options}{extension}` permet d'ajouter des options supplémentaires lorsqu'on chargera `extension` avec `\RequirePackage`.

Exemple pour passer toutes les options à geometry

Beamer3/code564.tex



Fichier

```
\DeclareOption*{%  
  \PassOptionsToPackage{\CurrentOption}{geometry}%  
}  
  
\ProcessOptions\relax  
  
\RequirePackage{geometry}
```

Commandes différées

- `\AtEndOfPackage{...}` exécute son argument à la fin de l'extension ;
- `\AtBeginDocument{...}` exécute son argument juste après `\begin{document}` ;
- `\AtEndDocument{...}` exécute son argument juste avant `\end{document}`.

Commandes différées

- `\AtEndOfPackage{...}` exécute son argument à la fin de l'extension ;
- `\AtBeginDocument{...}` exécute son argument juste après `\begin{document}` ;
- `\AtEndDocument{...}` exécute son argument juste avant `\end{document}`.

Exemple pour changer l'indentation des paragraphes

Beamer3/code566.tex



Fichier

```
\DeclareOption{superindent}{%  
  \AtBeginDocument{%  
    \setlength\parindent{1cm}%  
  }%  
}
```

Exercice

Faire une extension simple qui contient des commandes définies aujourd'hui

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée
 - Tests et branchements conditionnels
 - Écrire une nouvelle extension
 - **Écrire une nouvelle classe**
 - Définition avancée des commandes
 - Boucles
 - Codes de catégorie

Exemple minimaliste

Fichier maclasse.cls



Fichier Beamer3/code567.tex

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{maclasse}[2009/04/17_Ma_première_classe , V1.0]

%Base
\LoadClassWithOptions{article}

%Extensions
\RequirePackage{geometry}
```

Les bases

Le nom d'un fichier de classe se termine par l'extension `.cls`.

Les bases

Le nom d'un fichier de classe se termine par l'extension `.cls`.

Rien de particulier, à part :

- le fichier doit commencer par

Code



Fichier Beamer3/code569.tex

```
\NeedsTeXFormat{LaTeX2e}  
\ProvidesClass{nom}[Date>Description_courte,Version]
```

Les bases

Le nom d'un fichier de classe se termine par l'extension `.cls`.

Rien de particulier, à part :

- le fichier doit commencer par

Code



Fichier Beamer3/code570.tex

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{nom}[Date>Description_courte,Version]
```

- le caractère `@` n'est pas protégé ;

Le nom d'un fichier de classe se termine par l'extension `.cls`.

Rien de particulier, à part :

- le fichier doit commencer par

Code



Fichier Beamer3/code571.tex

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{nom}[Date>Description_courte,Version]
```

- le caractère `@` n'est pas protégé ;
- pour charger une autre extension on utilise non pas `\usepackage` mais `\RequirePackage[options]{extension}` ;

Le nom d'un fichier de classe se termine par l'extension `.cls`.

Rien de particulier, à part :

- le fichier doit commencer par

Code



Fichier Beamer3/code572.tex

```
\NeedsTeXFormat{LaTeX2e}  
\ProvidesClass{nom}[Date>Description_courte,Version]
```

- le caractère `@` n'est pas protégé ;
- pour charger une autre extension on utilise non pas `\usepackage` mais `\RequirePackage[options]{extension}` ;
- pour modifier une classe existante on utilise `\LoadClass[options]{classe existante}`.

Les options

Même principes que pour une extension, avec en plus :

- `\LoadClassWithOptions{classe existante}` au lieu de `\LoadClass` charge la classe existante avec exactement les mêmes options ;
- `\PassOptionsToClass{options}{classe existante}` ajoute une option qui sera activée au moment de `\loadclass`.

Les options

Même principes que pour une extension, avec en plus :

- `\LoadClassWithOptions{classe existante}` au lieu de `\LoadClass` charge la classe existante avec exactement les mêmes options ;
- `\PassOptionsToClass{options}{classe existante}` ajoute une option qui sera activée au moment de `\loadclass`.

Exemple pour passer toutes les options à `article`  Fichier Beamer3/code574.tex

```
%Si on ne déclare aucune option particulière, ce qui suit est  
équivalent à \LoadClassWithOptions{article}  
\DeclareOption*{%  
  \PassOptionsToClass{\CurrentOption}{article}%  
}  
  
\ProcessOptions\relax  
  
\LoadClass{article}
```

Exercice

Faire une classe simple dérivée de Beamer avec une option `article`

Solution

Faire une classe simple dérivée de Beamer avec une option article

Code



Fichier Beamer3/code575.tex

```
\newif\ifarticle
\DeclareOption{article}{\articlettrue}

\DeclareOption*{
  \ifarticle
  \PassOptionsToClass{\CurrentOption}{article}
  \else
  \PassOptionsToClass{\CurrentOption}{beamer}
  \fi
}

\ProcessOptions\relax

\ifarticle
\LoadClass{article}
\RequirePackage{beamerarticle}
\else
\LoadClass{beamer}
\fi
```

Exercice [Difficile]

Faire une classe simple pour écrire un CV

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 **Programmation \LaTeX avancée**
 - Tests et branchements conditionnels
 - Écrire une nouvelle extension
 - Écrire une nouvelle classe
 - **Définition avancée des commandes**
 - Boucles
 - Codes de catégorie

Définition de commandes

Les moyens propres à \LaTeX :

- `\newcommand\commande[n]{...}` définit une nouvelle `\commande` à `n` arguments (une erreur est déclenchée si `\commande` existe déjà, ou si son nom commence par `end`);

Définition de commandes

Les moyens propres à \LaTeX :

- `\newcommand\commande[n]{...}` définit une nouvelle `\commande` à `n` arguments (une erreur est déclenchée si `\commande` existe déjà, ou si son nom commence par `end`);
- `\providecommand\commande[n]{...}` ne définit `\commande` que si elle n'existe pas déjà;

Définition de commandes

Les moyens propres à \LaTeX :

- `\newcommand\commande[n]{...}` définit une nouvelle `\commande` à n arguments (une erreur est déclenchée si `\commande` existe déjà, ou si son nom commence par `end`);
- `\providecommand\commande[n]{...}` ne définit `\commande` que si elle n'existe pas déjà;
- `\renewcommand\commande[n]{...}` redéfinit une `\commande` existante (une erreur est déclenchée si `\commande` n'existe pas déjà);

Définition de commandes

Les moyens propres à \LaTeX :

- `\newcommand\commande[n]{...}` définit une nouvelle `\commande` à n arguments (une erreur est déclenchée si `\commande` existe déjà, ou si son nom commence par `end`);
- `\providcommand\commande[n]{...}` ne définit `\commande` que si elle n'existe pas déjà;
- `\renewcommand\commande[n]{...}` redéfinit une `\commande` existante (une erreur est déclenchée si `\commande` n'existe pas déjà);
- `\newenvironment{envi}[n]{avant}{après}` définit les deux commandes `\envi` et `\endenvi`, qui ont le même effet que `\begin{envi}` et `\end{envi}` à deux différences près :
 - ❶ pas d'effet de localisation propre aux groupements (retour aux couleurs et à la police précédente, ...),
 - ❷ pas de vérification d'imbrication correcte;

Définition de commandes

Les moyens propres à \LaTeX :

- `\newcommand\commande[n]{...}` définit une nouvelle `\commande` à n arguments (une erreur est déclenchée si `\commande` existe déjà, ou si son nom commence par `end`);
- `\providewcommand\commande[n]{...}` ne définit `\commande` que si elle n'existe pas déjà;
- `\renewcommand\commande[n]{...}` redéfinit une `\commande` existante (une erreur est déclenchée si `\commande` n'existe pas déjà);
- `\newenvironment{envi}[n]{avant}{après}` définit les deux commandes `\envi` et `\endenvi`, qui ont le même effet que `\begin{envi}` et `\end{envi}` à deux différences près :
 - ❶ pas d'effet de localisation propre aux groupements (retour aux couleurs et à la police précédente, ...),
 - ❷ pas de vérification d'imbrication correcte;
- `\renewenvironment{envi}[n]{avant}{après}` redéfinit `\envi` et `\endenvi`.

Définition de commandes

Les moyens propres à \LaTeX :

- `\newcommand\commande[n]{...}` définit une nouvelle `\commande` à n arguments (une erreur est déclenchée si `\commande` existe déjà, ou si son nom commence par `end`);
- `\providewcommand\commande[n]{...}` ne définit `\commande` que si elle n'existe pas déjà;
- `\renewcommand\commande[n]{...}` redéfinit une `\commande` existante (une erreur est déclenchée si `\commande` n'existe pas déjà);
- `\newenvironment{envi}[n]{avant}{après}` définit les deux commandes `\envi` et `\endenvi`, qui ont le même effet que `\begin{envi}` et `\end{envi}` à deux différences près :
 - ❶ pas d'effet de localisation propre aux groupements (retour aux couleurs et à la police précédente, ...),
 - ❷ pas de vérification d'imbrication correcte;
- `\renewenvironment{envi}[n]{avant}{après}` redéfinit `\envi` et `\endenvi`.

Toute (re)définition d'une `\commande` par l'un de ces moyens est globale et non limitée au groupe en cours.

Plusieurs arguments optionnels

`\newcommand\commande[n][valeur]{...}` rend le premier argument #1 optionnel, avec une valeur par défaut s'il n'est pas donné.

Plusieurs arguments optionnels

`\newcommand\commande[n][valeur]{...}` rend le premier argument #1 optionnel, avec une valeur par défaut s'il n'est pas donné.

On peut définir des commandes à plusieurs arguments optionnels à l'aide de définitions intermédiaires :

Code



Fichier Beamer3/code577.tex

```
%Le\begingroup/endgroup\sert\
seulement\à\revenir\à\la\couleur\
normale\à\la\fin\de\FLASHY
\newcommand\FLASHYAUX[2][yellow]{\
colorbox{#1}{#2}\endgroup}
\newcommand\FLASHY[1][blue]{\
begingroup\color{#1}\FLASHYAUX}

\FLASHY{Un\peu}

\FLASHY[violet]{Beaucoup}

\FLASHY[red][pink]{À\la\folie}
```

Sortie

Un peu

Beaucoup

À la folie

Variantes étoilées

`\@ifstar{étoile}{pas étoile}` teste si le prochain caractère est un caractère * (et le cas échéant, le fait « disparaître ») et si oui lit "étoile", sinon "pas étoile".

Variantes étoilées

`\@ifstar{étoile}{pas étoile}` teste si le prochain caractère est un caractère * (et le cas échéant, le fait « disparaître ») et si oui lit "étoile", sinon "pas étoile".

Code



Fichier Beamer3/code579.tex

```
\makeatletter
\newcommand\FRAMEA[1]{\fbox{#1}}
\newcommand\FRAMEB[1]{\ovalbox{#1}}
\newcommand\FRAME{\@ifstar{\FRAMEA}{\FRAMEB}}
```

```
Deux\FRAME{styles}\pour\FRAME*{
encadrer}.
```

Sortie

Deux styles pour encadrer.

Variantes étoilées

`\@ifstar{étoile}{pas étoile}` teste si le prochain caractère est un caractère * (et le cas échéant, le fait « disparaître ») et si oui lit "étoile", sinon "pas étoile".

Code



Fichier Beamer3/code580.tex

```
\makeatletter
\newcommand\FRAMEA [1]{\fbox{#1}}
\newcommand\FRAMEB [1]{\ovalbox{#1}}
\newcommand\FRAME{\@ifstar{\FRAMEA}{\FRAMEB}}
```

```
Deux\FRAME{styles}\pour\FRAME*{encadrer}.
```

Sortie

Deux styles pour encadrer.

Sur le même modèle :

- `\@ifnextchar{c}{égal}{pas égal}` permet de tester si le prochain jeton (caractère et code de catégorie) est égal à c ;
- `\@ifnextcharacter{c}{égal}{pas égal}` fait la même chose mais ne s'occupe pas des codes de catégorie.

Définition de macros

Les moyens propres à T_EX

`\def\macro#1#2...#n{...}` fait quasiment la même chose que
`\newcommand\macro[n]{...}` à quelques différences près :

- la définition est valable uniquement dans le bloc en cours (elle disparaît ensuite) ;
- pas de vérification de l'existence préalable de `\macro` ;
- la partie `#1#2...#n` autorise un « parsing » plus complexe.

Définition de macros

Les moyens propres à T_EX

`\def\macro#1#2...#n{...}` fait quasiment la même chose que
`\newcommand\macro[n]{...}` à quelques différences près :

- la définition est valable uniquement dans le bloc en cours (elle disparaît ensuite) ;
- pas de vérification de l'existence préalable de `\macro` ;
- la partie `#1#2...#n` autorise un « parsing » plus complexe.

Code



Fichier Beamer3/code582.tex

```
\def\TRUC#1{\textsc{#1}}
\def\TRUCDOUBLE(#1,#2){\textbf{#1}\textit{#2}}
\def\SUPERTRUC#1\par{{\red#1\par}}

\TRUC{Un argument standard}. \TRUCDOUBLE(Deux arguments, spéciaux). \SUPERTRUC Un argument qui va jusqu'à la fin du paragraphe en cours.
```

Prochain paragraphe...

Sortie

UN ARGUMENT STANDARD. **Deux arguments spéciaux.** Un argument qui va jusqu'à la fin du paragraphe en cours.

Prochain paragraphe...

Définition de macros

Les moyens propres à T_EX

`\let\macrob=\macroa` recopie la signification *exacte* de `\macroa` dans la séquence de contrôle `\macrob`.

- L'assignation est valable uniquement dans le bloc en cours (ensuite, tout redevient comme avant) ;
- pas de vérification de l'existence préalable de `\macroa` (auquel cas `\macrob` sera elle-même indéfinie) ;
- le signe égal est facultatif.

Définition de macros

Les moyens propres à T_EX

`\let\macrob=\macroa` recopie la signification *exacte* de `\macroa` dans la séquence de contrôle `\macrob`.

- L'assignation est valable uniquement dans le bloc en cours (ensuite, tout redevient comme avant) ;
- pas de vérification de l'existence préalable de `\macroa` (auquel cas `\macrob` sera elle-même indéfinie) ;
- le signe égal est facultatif.

Code



Fichier Beamer3/code584.tex

```
\let\bd=\description
\let\ed=\enddescription
\let\im\item
\bd
  \im[Un]deux
  \im[Trois]quatre
\ed
```

Sortie

Un deux
Trois quatre

Définition de macros

Les moyens propres à T_EX

`\edef\macro{texte}` va développer le `texte` (prise en compte des `\if...`, remplacement des macros par leur valeur...) pour en faire une liste de jetons de texte. Cette liste sera la nouvelle définition de la `\macro`.

- La définition est valable uniquement dans le bloc en cours ;
- c'est un moyen de développer des commandes sans les afficher ;
- la liste (après expansion) doit être composée de texte uniquement.

Définition de macros

Les moyens propres à T_EX

`\edef\macro{texte}` va développer le texte (prise en compte des `\if...`, remplacement des macros par leur valeur...) pour en faire une liste de jetons de texte. Cette liste sera la nouvelle définition de la `\macro`.

- La définition est valable uniquement dans le bloc en cours ;
- c'est un moyen de développer des commandes sans les afficher ;
- la liste (après expansion) doit être composée de texte uniquement.

Code



Fichier Beamer3/code586.tex

```
\def\listevide{}
\let\liste=\listevide
\def\ajouter#1{%
  \ifx\liste\listevide
    \edef\liste{#1}%
  \else
    \edef\liste{\liste , #1}%
  \fi
}

\ajouter{un}\ajouter{deux}\ajouter{
trois}\ajouter{quatre}
Voici la liste : \liste.
```

Sortie

Voici la liste : un, deux, trois, quatre.

Préfixes de définition

Il y en a trois, à placer avant `\def`, `\let`,
`\edef...` :

- `\global` rend la portée de la définition globale;

Préfixes de définition

Il y en a trois, à placer avant `\def`, `\let`,
`\edef...` :

- `\global` rend la portée de la définition globale ;
- `\outer` rend la portée de la définition localisée au groupe de niveau supérieur ;

Préfixes de définition

Il y en a trois, à placer avant `\def`, `\let`,
`\edef...` :

- `\global` rend la portée de la définition globale ;
- `\outer` rend la portée de la définition localisée au groupe de niveau supérieur ;
- `\long` avec `\def` autorise des définitions de macros dont les arguments peuvent contenir des sauts de paragraphe.

Préfixes de définition

Il y en a trois, à placer avant `\def`, `\let`,
`\edef...` :

- `\global` rend la portée de la définition globale ;
- `\outer` rend la portée de la définition localisée au groupe de niveau supérieur ;
- `\long` avec `\def` autorise des définitions de macros dont les arguments peuvent contenir des sauts de paragraphe.

Les commandes définies avec `\newcommand` sont automatiquement `\long`.

Préfixes de définition

Il y en a trois, à placer avant `\def`, `\let`, `\edef`... :

- `\global` rend la portée de la définition globale ;
- `\outer` rend la portée de la définition localisée au groupe de niveau supérieur ;
- `\long` avec `\def` autorise des définitions de macros dont les arguments peuvent contenir des sauts de paragraphe.

Les commandes définies avec `\newcommand` sont automatiquement `\long`.

Ces préfixes peuvent aussi s'utiliser avec les `\ifXXX` (`\global\XXXtrue`), avec `\addto` ou encore avec des registres de longueurs (`\global\parskip=1cm`).

Préfixes de définition

Il y en a trois, à placer avant `\def`, `\let`, `\edef`... :

- `\global` rend la portée de la définition globale ;
- `\outer` rend la portée de la définition localisée au groupe de niveau supérieur ;
- `\long` avec `\def` autorise des définitions de macros dont les arguments peuvent contenir des sauts de paragraphe.

Les commandes définies avec `\newcommand` sont automatiquement `\long`.

Ces préfixes peuvent aussi s'utiliser avec les `\ifXXX`

(`\global\XXXtrue`), avec `\addto` ou encore avec des registres de longueurs

(`\global\parskip=1cm`).

Il existe aussi des formes abrégées :

- `\gdef` pour `\global\def` ;
- `\ldef` pour `\long\def` ;
- `\xdef` pour `\global\edef`.

Appeler une commande par son nom

- `\csname texte\endcsname` désigne (après expansion de `texte`) la macro `\texte`;
- il faut utiliser `\expandafter` avant les définitions dans ce cas.

Appeler une commande par son nom

- `\csname texte\endcsname` désigne (après expansion de `texte`) la macro `\texte`;
- il faut utiliser `\expandafter` avant les définitions dans ce cas.

Code



Fichier Beamer3/code589.tex

```
\newcounter{moncompteur}
\newcommand\changeitemlabel[2]{%
  \setcounter{moncompteur}{#1}%
  \expandafter\def\csname\labelitem\
    roman{moncompteur}\endcsname{#2}%
}

\changeitemlabel{1}{\rule{1ex}{1ex}}
\begin{itemize}
  \item[\black\rule{1ex}{1ex}] Un
  \item[\black\rule{1ex}{1ex}] Deux
\end{itemize}
```

Sortie

- Un
- Deux

Déboggage

- `\show\commande` affiche la définition d'une macro ;
- `\showthe\registre` affiche la valeur d'un registre (une longueur par exemple).

Exercice

Définir une commande prenant un argument optionnel encadré par <...>

Solution

Définir une commande prenant un argument optionnel encadré par <...>

Code



Fichier Beamer3/code591.tex

```
\makeatletter
\def\machinARG<#1>{Avec␣argument :␣#1}
\def\machinNOARG{Sans␣argument}
\def\machin{\@ifnextchar<\machinARG\machinNOARG}
\machin<oui>.\␣\machin.
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée
 - Tests et branchements conditionnels
 - Écrire une nouvelle extension
 - Écrire une nouvelle classe
 - Définition avancée des commandes
 - **Boucles**
 - Codes de catégorie

Boucles "tant que" en T_EX

- `\@whilenum test\do{...}` répète son argument tant que la comparaison `test` entre nombres est vraie ;
- `\@whiledim test\do{...}` fait la même chose pour un test sur des longueurs.

Boucles "tant que" en T_EX

- `\@whilenum test\do{...}` répète son argument tant que la comparaison `test` entre nombres est vraie ;
- `\@whiledim test\do{...}` fait la même chose pour un test sur des longueurs.

Code



Fichier Beamer3/code594.tex

```
\newlength\LARGEUR
\newlength\LARGEURTOTALE
\makeatletter
\newcommand\REPLIR[2]{%
uu\settowidth\LARGEUR{#2}%
uu\setlength\LARGEURTOTALE{0pt}%
uu\@whiledim\LARGEURTOTALE<#1\do{%
uuuu#2%
uuuu\addtolength\LARGEURTOTALE{\
LARGEUR}%
uu}%
}

\REPLIR{3cm}{+}\REPLIR{3cm}{.}\REPLIR{3cm}{\#}
```

Sortie

```
+++++
.....
#####
```

Boucles "tant que" en \LaTeX

Avec l'extension `ifthen`

`\whiledo{condition}{...}` répète son second argument tant que la condition (décrite avec les mêmes conventions que `\ifthenelse`) est vraie.

Boucles "tant que" en \LaTeX

Avec l'extension `ifthen`

`\whiledo{condition}{...}` répète son second argument tant que la condition (décrite avec les mêmes conventions que `\ifthenelse`) est vraie.

Code



Fichier `Beamer3/code598.tex`

```
\newcounter{ca}\newcounter{cb}
\newcommand\pgcd[2]{%
uu\setcounter{ca}{#1}%
uu\setcounter{cb}{#2}%
uu\pgcd(#1,#2)\begin{array}[t]{cl}%
uu\whiledo{\not\(\value{ca}=\value{cb}
)\)}{%
uuuu\ifthenelse{\value{ca}>\value{cb}
}{%
uuuuuu\addtocounter{ca}{-\value{cb}}%
uuuuu}%
uuuuuu\addtocounter{cb}{-\value{ca}}%
uuuuu}%
uuuu=&\pgcd(\theca,\thecb)\
uu}%
uu=&\theca.
uu\end{array}
}
```

`\[\pgcd{512}{388}\]`

Sortie

```
pgcd(512, 388)=pgcd(124, 388)
=pgcd(124, 264)
=pgcd(124, 140)
=pgcd(124, 16)
=pgcd(108, 16)
=pgcd(92, 16)
=pgcd(76, 16)
=pgcd(60, 16)
=pgcd(44, 16)
=pgcd(28, 16)
=pgcd(12, 16)
=pgcd(12, 4)
=pgcd(8, 4)
=pgcd(4, 4)
=4.
```

Boucles "pour" en T_EX

Attention au code de catégorie de ':' qui est changé par french ou [français]{babel}!

- `\@for\var:=liste\do{...}` répète son argument avec `\var` qui parcourt les éléments de la `liste`. Les éléments de la liste doivent être séparés par des virgules;
- `\@tfor\var:=texte\do{...}` fait la même chose, en parcourant toutes les lettres du `texte`.

Boucles "pour" en T_EX

Attention au code de catégorie de ':' qui est changé par french ou [français]{babel}!

- `\@for\var:=liste\do{...}` répète son argument avec `\var` qui parcourt les éléments de la liste. Les éléments de la liste doivent être séparés par des virgules;
- `\@tfor\var:=texte\do{...}` fait la même chose, en parcourant toutes les lettres du texte.

Code



Fichier Beamer3/code601.tex

```
\begin{itemize}
\catcode '\:=12
\makeatletter
\@for\x:=1,2,3,soleil!\do{\item\x}
\end{itemize}
```

Sortie

- 1
- 2
- 3
- soleil!

Exercice

Écrire 100 fois « Je n'utiliserai pas les boucles en cours de \LaTeX »

Solution

Écrire 100 fois « Je n'utiliserai pas les boucles en cours de L^AT_EX »

Code



Fichier Beamer3/code602.tex

```
\newcounter{puniton}  
\makeatletter  
\@whilenum\thepuniton<100\do{%  
  Je n'utiliserai pas les boucles en cours de LATEX\\%  
  \stepcounter{puniton}%  
}
```

Exercice

Écrire une commande `\vertical` qui permette d'afficher verticalement son argument dans un tableau

Solution

Écrire une commande `\vertical` qui permette d'afficher verticalement son argument dans un tableau

Code



Fichier Beamer3/code603.tex

```
\makeatletter
\catcode'\:=12

\newcommand\VERTICAL [1]{%
  \begin{tabular}[t]{c}%
    \@tfor\XXX:=#1\do{%
      \XXX\\%
    }%
  \end{tabular}%
}
```

Table des matières

- 1 Types de documents supplémentaires en \LaTeX
- 2 Utiliser \LaTeX pour le web
- 3 Trucs et astuces avancés
- 4 Programmation \LaTeX avancée
 - Tests et branchements conditionnels
 - Écrire une nouvelle extension
 - Écrire une nouvelle classe
 - Définition avancée des commandes
 - Boucles
 - Codes de catégorie

Les "tokens" et leurs "catcodes"

En première lecture, L^AT_EX transforme le texte en une liste de jetons (tokens) :

- soit des "caractères" (table ASCII) ;
- soit des "séquences de contrôle" (noms de commandes).

Code



Fichier Beamer3/code604.tex

```
2 est \textit{premier}.
```

```
%Commentaire
```

Jetons

```
2 est textit { p r e m i e r } . par par
```

Les "tokens" et leurs "catcodes"

En plus de sa valeur en tant que caractère, chaque jeton qui n'est pas une séquence de contrôle possède un attribut supplémentaire compris entre 0 et 15 : le code de catégorie (catcode).

Code



Fichier Beamer3/code605.tex

```
2 est \textit{premier}.
```

%Commentaire

Jetons

```
212 ω10 e11 s11 t11 ω10 textit {1 p11 r11 e11 m11 i11 e11 r11 }2 ·12 ω5 par par
```

Codes de catégorie par défaut

0 – caractère d'échappement (début
d'une séquence de contrôle) : '\'

Codes de catégorie par défaut

- 0 – caractère d'échappement : `'\'`
- 1 – délimiteur gauche : `'{'`
- 2 – délimiteur droit : `'}'`
- 3 – délimiteur mathématique : `'$'`

Codes de catégorie par défaut

- 0 – caractère d'échappement : '\'
- 1 – délimiteur gauche : '{'
- 2 – délimiteur droit : '}'
- 3 – délimiteur mathématique : '\$'
- 4 – tabulation : '&'
- 5 – fin de ligne : '^M'

Codes de catégorie par défaut

- 0 – caractère d'échappement : '\'
- 1 – délimiteur gauche : '{'
- 2 – délimiteur droit : '}'
- 3 – délimiteur mathématique : '\$'
- 4 – tabulation : '&'
- 5 – fin de ligne : '^M'
- 6 – paramètre de macro : '#'

Codes de catégorie par défaut

0 – caractère d'échappement : '\'

1 – délimiteur gauche : '{'

2 – délimiteur droit : '}'

3 – délimiteur mathématique : '\$'

4 – tabulation : '&'

5 – fin de ligne : '^M'

6 – paramètre de macro : '#'

7 – exposant : '^'

8 – indice : '_'

Codes de catégorie par défaut

0 – caractère d'échappement : '\'

1 – délimiteur gauche : '{'

2 – délimiteur droit : '}'

3 – délimiteur mathématique : '\$'

4 – tabulation : '&'

5 – fin de ligne : '^M'

6 – paramètre de macro : '#'

7 – exposant : '^'

8 – indice : '_'

9 – caractères ignorés : '^00'

Codes de catégorie par défaut

- 0 – caractère d'échappement : `'\'`
- 1 – délimiteur gauche : `'{'`
- 2 – délimiteur droit : `'}'`
- 3 – délimiteur mathématique : `'$'`
- 4 – tabulation : `'&'`
- 5 – fin de ligne : `'^M'`
- 6 – paramètre de macro : `'#'`
- 7 – exposant : `'^'`

- 8 – indice : `'_'`
- 9 – caractères ignorés : `'^00'`
- 10 – espaces : `'\u00A0', '\u00A1'`

Codes de catégorie par défaut

- 0 – caractère d'échappement : `'\'`
- 1 – délimiteur gauche : `'{'`
- 2 – délimiteur droit : `'}'`
- 3 – délimiteur mathématique : `'$'`
- 4 – tabulation : `'&'`
- 5 – fin de ligne : `'^M'`
- 6 – paramètre de macro : `'#'`
- 7 – exposant : `'^'`

- 8 – indice : `'_'`
- 9 – caractères ignorés : `'^^00'`
- 10 – espaces : `'\u00a0'`, `'\u00a1'`
- 11 – lettres : `'a'`, `'z'`, `'E'...`

Codes de catégorie par défaut

- 0 – caractère d'échappement : '\'
- 1 – délimiteur gauche : '{'
- 2 – délimiteur droit : '}'
- 3 – délimiteur mathématique : '\$'
- 4 – tabulation : '&'
- 5 – fin de ligne : '^M'
- 6 – paramètre de macro : '#'
- 7 – exposant : '^'

- 8 – indice : '_'
- 9 – caractères ignorés : '^00'
- 10 – espaces : ' ', '^I'
- 11 – lettres (peuvent former des séquences de contrôle à plusieurs caractères) : 'a', 'z', 'E'...
- 12 – autres caractères : '@', '1', '('...

Codes de catégorie par défaut

- 0 – caractère d'échappement : '\'
- 1 – délimiteur gauche : '{'
- 2 – délimiteur droit : '}'
- 3 – délimiteur mathématique : '\$'
- 4 – tabulation : '&'
- 5 – fin de ligne : '^M'
- 6 – paramètre de macro : '#'
- 7 – exposant : '^'

- 8 – indice : '_'
- 9 – caractères ignorés : '^00'
- 10 – espaces : ' ', '^I'
- 11 – lettres : 'a', 'z', 'E'...
- 12 – autres caractères (tous ceux qui ne sont pas dans cette liste) : '@', '1', '('...
- 13 – caractères actifs : '~'

Codes de catégorie par défaut

- 0 – caractère d'échappement : `'\'`
- 1 – délimiteur gauche : `'{'`
- 2 – délimiteur droit : `'}'`
- 3 – délimiteur mathématique : `'$'`
- 4 – tabulation : `'&'`
- 5 – fin de ligne : `'^M'`
- 6 – paramètre de macro : `'#'`
- 7 – exposant : `'^'`
- 8 – indice : `'_'`
- 9 – caractères ignorés : `'^00'`
- 10 – espaces : `'\u0020'`, `'\u00A0'`
- 11 – lettres : `'a'`, `'z'`, `'E'...`
- 12 – autres caractères : `'@'`, `'1'`, `'('...`
- 13 – caractères actifs (peuvent être utilisés comme une séquence de contrôle) : `'~'`
- 14 – début d'un commentaire : `'%'`
- 15 – caractères invalides.

Changer les codes de catégorie

Pour faire passer le code de catégorie des prochains caractères α à la valeur n dans le groupe en cours :

```
\catcode'\alpha=n
```

Changer les codes de catégorie

Pour faire passer le code de catégorie des prochains caractères α à la valeur n dans le groupe en cours :

$$\backslash\text{catcode}\alpha=n$$

Par exemple :

- `\makeatletter` change le code de catégorie de @ à 11 (`\catcode'\@=11`);
- `\makeatother` change le code de catégorie de @ à 12 (`\catcode'\@=12`).

Changer les codes de catégorie

Pour faire passer le code de catégorie des prochains caractères α à la valeur n dans le groupe en cours :

$$\backslash\text{catcode}'\alpha=n$$

Par exemple :

- `\makeatletter` change le code de catégorie de @ à 11 (`\catcode'\@=11`);
- `\makeatother` change le code de catégorie de @ à 12 (`\catcode'\@=12`).

Code



Fichier Beamer3/code608.tex

```
\catcode'\_ =11
\newcommand\super-commande{Super_
  commande!_}

\super-commande\super-commande\super-
  commande
```

Sortie

Super commande! Super commande! Su-
per commande!

La catégorie 13 (caractères actifs)

Les caractères de cette catégorie peuvent être utilisés comme une séquence de contrôle.

La catégorie 13 (caractères actifs)

Les caractères de cette catégorie peuvent être utilisés comme une séquence de contrôle.

Par exemple, les extensions `french` ou `[français]{babel}` rendent les caractères de ponctuation doubles (`:` `;` `!` `?`) actifs pour gérer correctement les espaces à la française.

La catégorie 13 (caractères actifs)

Les caractères de cette catégorie peuvent être utilisés comme une séquence de contrôle.

Par exemple, les extensions `french` ou `[français]{babel}` rendent les caractères de ponctuation doubles (`:` `;` `!` `?`) actifs pour gérer correctement les espaces à la française.

Code



Fichier Beamer3/code611.tex

```
Le @ par défaut est moche?
```

```
\catcode '\@=13  
\newcommand{\MVAt}  
Utiliser le @ de l'extension marvosym  
!
```

Sortie

Le @ par défaut est moche?

Utiliser le @ de l'extension marvosym!

Règle d'or des codes de catégories

Une fois qu'un caractère a été lu, son code de catégorie ne peut plus être changé.

Règle d'or des codes de catégories

Une fois qu'un caractère a été lu, son code de catégorie ne peut plus être changé.

Pourquoi l'utilisation de `\letitre` va-t-elle provoquer une erreur ?

 Fichier Beamer3/code613.tex

```
\newcommand\letitre{%  
  \makeatletter  
  \@title%  
  \makeatother  
}
```

Fonctionnement (simplifié) de la commande `\verb`

- 1 les codes de catégorie de tous les caractères spéciaux (à part le saut de ligne) sont changés en 12 (autre);

Fonctionnement (simplifié) de la commande `\verb`

- 1 les codes de catégorie de tous les caractères spéciaux (à part le saut de ligne) sont changés en 12 (autre);
- 2 on commence un groupe;

Fonctionnement (simplifié) de la commande `\verb`

- 1 les codes de catégorie de tous les caractères spéciaux (à part le saut de ligne) sont changés en 12 (autre) ;
- 2 on commence un groupe ;
- 3 le premier caractère qui suit (α) est lu ;

Fonctionnement (simplifié) de la commande `\verb`

- 1 les codes de catégorie de tous les caractères spéciaux (à part le saut de ligne) sont changés en 12 (autre);
- 2 on commence un groupe;
- 3 le premier caractère qui suit (α) est lu;
- 4 son code de catégorie est changé en 13 (actif);

Fonctionnement (simplifié) de la commande `\verb`

- 1 les codes de catégorie de tous les caractères spéciaux (à part le saut de ligne) sont changés en 12 (autre) ;
- 2 on commence un groupe ;
- 3 le premier caractère qui suit (α) est lu ;
- 4 son code de catégorie est changé en 13 (actif) ;
- 5 la commande associée à α termine le groupe (et donc restaure les catégories initiales).

Fonctionnement (simplifié) de la commande `\verb`

- 1 les codes de catégorie de tous les caractères spéciaux (à part le saut de ligne) sont changés en 12 (autre) ;
- 2 on commence un groupe ;
- 3 le premier caractère qui suit (α) est lu ;
- 4 son code de catégorie est changé en 13 (actif) ;
- 5 la commande associée à α termine le groupe (et donc restaures les catégories initiales).

Pourquoi ce code ne produit-il pas d'erreur, et qu'affiche-t-il ?



Fichier

Beamer3/code619.tex

```
\newcommand\test [1]{\verb+#1+}
```

```
\test{Oui}Non+
```

Fonctionnement (simplifié) de la commande `\verb`

- 1 les codes de catégorie de tous les caractères spéciaux (à part le saut de ligne) sont changés en 12 (autre) ;
- 2 on commence un groupe ;
- 3 le premier caractère qui suit (α) est lu ;
- 4 son code de catégorie est changé en 13 (actif) ;
- 5 la commande associée à α termine le groupe (et donc restaure les catégories initiales).

Pourquoi ce code ne produit-il pas d'erreur, et qu'affiche-t-il ?  Fichier

Beamer3/code620.tex

```
\newcommand\test[1]{\verb+#1+}
```

```
\test{Oui}Non+
```

Réponse

```
Oui+Non
```

Significations spéciales du caractère '^'

Lorsqu'il est doublé, il permet d'insérer des caractères sans les taper explicitement :

- ^^xy est lu comme un seul caractère dont le code ASCII est xy en hexadécimal ;

Significations spéciales du caractère '^'

Lorsqu'il est doublé, il permet d'insérer des caractères sans les taper explicitement :

- $\text{^^}xy$ est lu comme un seul caractère dont le code ASCII est xy en hexadécimal ;
- $\text{^^}\alpha$ est lu comme un seul caractère β en appliquant la règle suivante :
 - soit n le code ASCII de α ;
 - si $64 \leq n \leq 127$ alors β a pour code ASCII $n - 64$;
 - si $0 \leq n \leq 63$ alors β a pour code ASCII $n + 64$.

Par exemple, $\text{^^}M$ désigne le caractère de saut de ligne, $\text{^^}I$ le caractère de tabulation du clavier.

Significations spéciales du caractère '^'

Lorsqu'il est doublé, il permet d'insérer des caractères sans les taper explicitement :

- $\text{\^{}xy}$ est lu comme un seul caractère dont le code ASCII est xy en hexadécimal ;
- $\text{\^{} α }$ est lu comme un seul caractère β en appliquant la règle suivante :
 - soit n le code ASCII de α ;
 - si $64 \leq n \leq 127$ alors β a pour code ASCII $n - 64$;
 - si $0 \leq n \leq 63$ alors β a pour code ASCII $n + 64$.

Par exemple, $\text{\^{}M}$ désigne le caractère de saut de ligne, $\text{\^{}I}$ le caractère de tabulation du clavier.

Code



Fichier Beamer3/code623.tex

```
\itshape_Pas_facile_de_taper_rapidement!  
  
%Pas_de_chance,_la_touche_e_(code_ascii_65_en_hexadécimal)_de_mon_clavier_vient_de_se_casser...  
\itshap65_Pas_facil65d65tap65r65rapid65m65%nt!
```

Sortie

```
Pas facile de taper rapidement!  
Pas facile de taper rapidement!
```

Exercice

Donner un code \LaTeX permettant de taper directement le symbole de l'euro, par exemple en se basant sur le symbole `\texteuro` de l'extension `textcomp`

Solution

Donner un code \LaTeX permettant de taper directement le symbole de l'euro, par exemple en se basant sur le symbole `\texteuro` de l'extension `textcomp`

Code



Fichier Beamer3/code624.tex

```
\catcode'\e=13
\newcommand{\texteuro}
%On peut aussi utiliser les symboles \EUR ou \EURtm de l'extension
marvosym
```

Exercice

Comment obliger \LaTeX à respecter les mêmes sauts de ligne que ceux tapés dans le document, au lieu de les transformer en espaces ?

Solution

Comment obliger \LaTeX à respecter les mêmes sauts de ligne que ceux tapés dans le document, au lieu de les transformer en espaces ?

Code



Fichier Beamer3/code625.tex

```
\catcode'\^^M=13%  
\def^^M{\ifhmode\newline\fi}%
```

Exercice [Difficile]

Donner un code \LaTeX permettant d'afficher en rouge les commentaires d'un document

Solution

Donner un code \LaTeX permettant d'afficher en rouge les commentaires d'un document

Code



Fichier Beamer3/code626.tex

```
\def\makeEOLother{\catcode'\^^M=12\relax}
{%
  \makeEOLother%
  \gdef\makeEOLspace{\catcode'\^^M=5\relax}%
  \gdef\eatline#1^^M{%
    \color{red}\_#1\_}%
    \makeEOLspace%
    \makePERCENTactive%
  }%
}

\def\makePERCENTactive{\catcode\%=13\relax}
\makePERCENTactive
\def\makePERCENTother{\catcode'\%=12\relax}
\def%\{\makeEOLother\makePERCENTother\eatline}
```