

Modèles \LaTeX

Recueil de templates

Tableaux, couleurs, styles,...



Table des matières

1 • Tableaux	II
1.1 Coloration des lignes et colonnes	III
1.1.1 Packages utilisés	III
1.1.2 Coloration de lignes	III
1.1.3 Coloration de colonnes	III
1.1.4 Coloration de cellules	III
2 • Équations	IV
2.1 Coloration des lignes et colonnes	V
2.1.1 Packages utilisés	V
2.1.2 Coloration de lignes	V
2.1.3 Coloration de colonnes	V
2.1.4 Coloration de cellules	V
3 • Polices	VI
3.1 Utilisation des polices sous XeLaTeX	VII
3.1.1 Indiquer la police par défaut du document	VII
3.1.2 Utiliser une police localement	VII
3.1.3 Les possibilités des polices	VII
3.1.4 Trouver les noms des polices	VIII
3.1.5 Restreindre le périmètre d'utilisation d'une police	VIII
3.1.6 Quelques exemples	VIII
4 • Fichiers	IX
4.1 Inclure des PDF	I
4.1.1 Packages utilisés	I
4.1.2 Exemple de commandes	I
4.1.3 Retourner les pages	I
4.1.4 Pour aller plus loin	I

1 • Tableaux

1.1 Coloration des lignes et colonnes

1.1.1 Packages utilisés

À minima, utiliser le package `colortbl` pour ajouter de la couleur aux lignes et colonnes. L'utilisation du package `xcolor` est recommandée pour faciliter la création de couleurs personnalisées.

```
1 \usepackage{xcolor} % Pour définir des couleurs personnalisées
2 \usepackage{colortbl} % Pour réaliser les différentes colorations
```

Pour les exemples qui suivent, nous retiendrons par exemple les définitions de couleurs suivantes :

```
1 \definecolor{Gray}{gray}{0.85}
2 \definecolor{LightCyan}{rgb}{0.88,1,1}
```

1.1.2 Coloration de lignes

```
1 \begin{table}
2   \begin{tabular}{c | c | c | c | c}
3     \hline
4     \rowcolor{LightCyan}
5     & x & y & w & z \\
6     \hline
7     variable 1 & a & b & c & d \\
8     variable 2 & a & b & c & d \\
9     \hline
10    \end{tabular}
11  \end{table}
```

	x	y	w	z
variable 1	a	b	c	d
variable 2	a	b	c	d

1.1.3 Coloration de colonnes

```
1 \newcolumntype{a}{>\columncolor{Gray}c}
2 \newcolumntype{b}{>\columncolor{white}c}
3
4 \begin{table}
5   \begin{tabular}{c | a | b | a | b}
6     \hline
7     & x & y & w & z \\
8     \hline
9     variable 1 & a & b & c & d \\
10    variable 2 & a & b & c & d \\
11    \hline
12    \end{tabular}
13  \end{table}
```

	x	y	w	z
variable 1	a	b	c	d
variable 2	a	b	c	d

1.1.4 Coloration de cellules

```
1 \begin{table}
2   \begin{tabular}{c | c | c | c | c}
3     \hline
4     & x & y & w & z \\
5     \hline
6     variable 1 & a & \cellcolor{Gray}b & c & d \\
7     variable 2 & a & b & c & d \\
8     \hline
9     \end{tabular}
10  \end{table}
```

	x	y	w	z
variable 1	a	b	c	d
variable 2	a	b	c	d

2 • Équations

2.1 Coloration des lignes et colonnes

2.1.1 Packages utilisés

Les commandes de base sont fournies par le package `colortbl`. Il est cependant possible de ne charger que les commandes pour tableaux via `xcolor`.

```
1 \usepackage[table]{xcolor} % [table] pour les options de coloration de tableaux
```

2.1.2 Coloration de lignes

```
1 \left(
2   \begin{array}{cccc}
3     \rowcolor{red!20}
4     x & x & x & x \\
5     0 & x & x & x \\
6     \rowcolor{blue!20}
7     0 & 0 & x & x \\
8     0 & 0 & 0 & x \\
9   \end{array}
10  \right)
```

$$\begin{pmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{pmatrix}$$

2.1.3 Coloration de colonnes

```
1 \left(
2   \begin{array}{>{\columncolor{olive!20}}cc>{\columncolor{orange!20}}cc}
3     x & x & x & x \\
4     0 & x & x & x \\
5     0 & 0 & x & x \\
6     0 & 0 & 0 & x \\
7   \end{array}
8   \right)
```

$$\begin{pmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{pmatrix}$$

2.1.4 Coloration de cellules

```
1 \newcommand\cc{\cellcolor{green!20}}
2
3 \left(
4   \begin{array}{cccc}
5     \cc x & x & x & x \\
6     0 & \cc x & x & x \\
7     0 & 0 & \cc x & x \\
8     0 & 0 & 0 & \cc x \\
9   \end{array}
10  \right)
```

$$\begin{pmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{pmatrix}$$

3 • Polices

3.1 Utilisation des polices sous XeLaTeX

L'apparition de XeLaTeX et LuaLaTeX a simplifié l'utilisation des polices pour les utilisateurs de \LaTeX , dans la mesure où toute police installée sur le système peut être utilisée par ces compilateurs en utilisant le package `fontspec` pour charger la police.

Par défaut, XeLaTeX et LuaLaTeX suppose que les textes sont écrits en UTF-8, il ne faut donc pas charger le package `inputenc` en les utilisant, et les fichiers sources peuvent être sauvegardés en UTF-8.

Ainsi, un document de base aura l'aspect suivant :

```
1 \documentclass{article}
2
3 \usepackage{fontspec}
4 \setmainfont{<any font on your system>}
5
6 \begin{document}
7 ...
8 \end{document}
```

3.1.1 Indiquer la police par défaut du document

Pour un document entier, on peut spécifier :

- ⊙ La police principale en style « roman ».
- ⊙ La police principale pour le style « sans empattement ».
- ⊙ La police principale pour le style « largeur fixe ».

Les commandes à utiliser sont les suivantes :

```
1 \setmainfont[<font features>]{<name of font>} % police roman
2 \setsansfont[<font features>]{<name of font>} % police sans empattements
3 \setmonofont[<font features>]{<name of font>} % police avec largeur fixe
```

3.1.2 Utiliser une police localement

Pour l'utilisation d'une police dans une partie précise du document, le mieux est de définir une nouvelle famille de polices.

```
1 \newfontfamily\myfont[<font features>]{<name of font>}
```

Ceci crée un déclencheur pour la police, appelé `\myfont` qui permet de passer à la police désignée. Il est également possible de sélectionner directement une fonte avec la commande `\fontspec`, cependant ceci n'est pas conseillé, dans la mesure où la méthode `\newfontfamily` est bien plus efficace.

3.1.3 Les possibilités des polices

Comme le package `fontspec` fournit des accès aux polices de type Open Type, il est possible d'accéder à des nombreuses fonctionnalités particulières de ces polices. Ces fonctions peuvent être sélectionnées en utilisant les arguments optionnels pour la sélection de n'importe quelle police. Pour plus de détails, se référer à la documentation du package `fontspec`, les plus courantes sont :

- ⊙ `[Ligatures=TeX]` Cela permet d'utiliser les ligatures \TeX usuelles (qui ne sont pas activées par défaut dans `fontspec`, ceci est particulièrement utiles par exemple pour saisir les tirets de citation – et – au lieu de saisir les vrais caractères directement). Cette option devrait toujours être utilisée.
- ⊙ `[Numbers=OldStyle]` Ceci permet d'avoir les version minuscules des chiffres.
- ⊙ `[Scale=MatchLowercase]` Ceci permet de mettre à dimension la police. Par exemple, les polices sans empattements et à largeur fixe pourront ainsi avoir des dimensions similaires à la version roman. Une autre option possible est `MatchUppercase`; il est aussi possible d'utiliser une valeur numérique pour la mise à l'échelle.

Si vous définissez de manière séparée une police roman, largeur fixe et sans empattements (ou si vous utilisez différentes familles de polices), il peut être utile de spécifier des paramètres pour toutes les polices. Ceci peut être exécuté avec la commande `\defaultfontfeatures` :

```
1 \defaultfontfeatures{Ligatures=TeX} % application à toutes les styles
   personnalisés
```

3.1.4 Trouver les noms des polices

Si vous désirez utiliser les polices installées sur votre système, vous pouvez utiliser le nom de la police tel qu'il apparaît dans toutes les applications de votre système. Pour Windows, ces polices sont situées dans `\Windows\Fonts`; pour MAC, elles sont dans `/Library/Fonts` et pour Linux elles sont dans `/usr/local/share/fonts`. Par exemple :

```
1 \setmainfont{Linux Libertine O}
```

À noter que si une police présente un espace dans son nom, il doit également être utilisé quand on décide l'utiliser avec `fontspec`.

3.1.5 Restreindre le périmètre d'utilisation d'une police

Il est toujours possible de restreindre la partie du document sur laquelle on réalise un changement de police. Pour cela, on utilise des groupes qui sont définis par des parenthèses :

```
1 {\myfont ...}
```

On peut également (pour plus de visibilité), définir des groupes sans les accolades :

```
1 \begingroup
2   \myfont ...
3 \endgroup
```

Si vous pensez utiliser souvent cette fonctionnalité, on peut même créer un environnement qui le permet :

```
1 \newenvironment{myfont}{\myfont}{\par}
2
3 \begin{myfont}
4   Some text in the new font.
5 \end{myfont}
```

Il est enfin possible de définir une commande pour standardiser le changement de polices, de la même manière que `\textrm` ou `\textbackslash textsf`, mais avec une police personnalisée :

```
1 \DeclareTextFontCommand{\textmyfont}{\myfont}
2
3 Texte avec police par défaut. \textmyfont{Nouvelle police.} De nouveau celle
   par défaut.
```

3.1.6 Quelques exemples

```
1 \usepackage{fontspec} % à mettre avant tout
   autre package sur les polices
2
3 \newfontfamily\aniron{Aniron}
4 \newfontfamily\arial{Arial}
5 \newfontfamily\arcena{AR CENA}
6
7 \begin{document}
8   {\arial Bonjour à la police Arial.}
9   {\aniron Bonjour à la police Aniron.}
10  {\arcena Bonjour à la police AR Cena Moyen.}
11 \end{document}
```

Test police Arial.
TEST police ANIRON.
 Test police AR Cena.

4 • Fichiers

4.1 Inclure des PDF

4.1.1 Packages utilisés

Le package recommandé ici est `pdfpages`, qui permet d'insérer des pages d'un PDF (éventuellement le document entier), selon des dispositions précisées par l'utilisateur.

```
1 \usepackage{pdfpages} % Pour inclure des pages d'un PDF
```

4.1.2 Exemple de commandes

Nous ne détaillerons pas outre mesure le fonctionnement du package pour le moment. Voici cependant dans un premier temps pour l'insertion de toutes les pages d'un document selon une grille 2×3 :

```
1 \includepdf[pages=-, nup=2x3]{chemin_relatif_vers_pdf}
```

On notera ici le `pages=-` qui permet d'insérer toutes les pages du documents. Si nous avions (pas exemple) voulu insérer que les pages 10 à 20, nous aurions pu écrire :

```
1 \includepdf[pages=10-20, nup=2x3]{chemin_relatif_vers_pdf}
```

4.1.3 Retourner les pages

Si vous avez pas exemple besoin d'effectuer une rotation sur les pages que vous insérez, vous pouvez utiliser l'option `angle`, comme dans cet exemple :

```
1 \includepdf[pages=10-20, nup=2x3, angle=-90]{chemin_relatif_vers_pdf}
```

4.1.4 Pour aller plus loin

Si vous désirez en savoir plus, rendez-vous à la documentation de ce package, à l'adresse suivante :

<http://www.texdoc.net/texmf-dist/doc/latex/pdfpages/pdfpages.pdf>

5 • Utilisation de Beamer

5.1 Créer son propre thème Beamer

5.1.1 Généralités

À titre de démonstration, un utilisateur de `tex.stackexchange` fournit un exemple de création de template pour Beamer en une heure. Cet exemple peut servir de base à des constructions plus complexes.

De nombreuses améliorations peuvent être apportées au code proposé, qui l'est ici à titre d'exemple. Pour l'exemple, le nouveau thème créé sera nommé `texsx`. Pour cela, on commence par créer les fichiers suivants dans le répertoire du thème :

- ⊙ `beamercolorthemetexsx.sty`
- ⊙ `beamerinnerthemetexsx.sty`
- ⊙ `beamerouterthemetexsx.sty`
- ⊙ `beamerthemetexsx.sty`
- ⊙ `test.tex` Pour tester le template créé.

Pour créer ce template, nous pourrions par exemple utiliser TikZ, tant pour sa puissance que pour sa rapidité d'exécution.

5.1.2 Objectif

Le but est de réussir à compiler un document pour Beamer de la forme suivante, avec page de titre, blocs, listes,...

```

1 \documentclass{beamer}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \title{There Is No Largest Prime Number}
5 \date[ISPN 80]{27th International Symposium of Prime Numbers}
6 \author[Euclid]{Euclid of Alexandria \texttt{euclid@alexandria.edu}}
7
8 \usetheme{texsx}
9
10 \begin{document}
11
12 \begin{frame}
13   \titlepage
14 \end{frame}
15
16
17 \begin{frame}
18   \frametitle{There Is No Largest Prime Number}
19   \framesubtitle{The proof uses \textit{reductio ad absurdum}.}
20   \begin{theorem}
21     There is no largest prime number. \end{theorem}
22   \begin{enumerate}
23     \item<1-| alert@1> Suppose  $p$  were the largest prime number.
24     \item<2-> Let  $q$  be the product of the first  $p$  numbers.
25     \item<3-> Then  $q+1$  is not divisible by any of them.
26     \item<1-> But  $q + 1$  is greater than  $1$ , thus divisible by some prime
27       number not in the first  $p$  numbers.
28   \end{enumerate}
29 \end{frame}
30
31 \begin{frame}{A longer title}
32   \begin{itemize}
33     \item one
34     \item two
35   \end{itemize}
36 \end{frame}

```

```
37
38 \end{document}
```

5.1.3 Fichier *beamerthemetexsx.sty*

Ce fichier permet de charger les fichiers de thèmes, et de définir certaines configurations générales (comme les symboles de navigation), ou de configurer en partie les blocs d’affichage. Un exemple vous est fourni ci-dessous :

```
1 \mode<presentation>
2
3 % Requirement
4 \RequirePackage{tikz}
5
6 % Settings
7 \useinnertheme{texsx}
8 \useoutertheme{texsx}
9 \usecolortheme{texsx}
10
11 \setbeamertemplate{navigation symbols}{}
12 \setbeamertemplate{blocks}[rounded][shadow=true]
13
14 \mode<all>
```

5.1.4 Slide de titre

Le style pour cette slide est défini dans le fichier *inner*. L’idée est ici d’exploiter les éléments suivants :

- ⊙ Utiliser le template *background* pour le fond de la slide de titre.
- ⊙ Définir un template *title page* dans lequel le titre, l’auteur et la date sont remplis.

D’un point de vue pratique, ces opérations se concrétisent avec le code suivant à insérer dans le fichier *beamerinnerthemetexsx.sty* :

```
1 \mode<presentation>
2
3 \setbeamertemplate{background}{
4     \begin{tikzpicture}
5         \useasboundingbox (0,0) rectangle(\the\paperwidth,\the\paperheight);
6         \fill[color=brown] (0,2) rectangle(\the\paperwidth,\the\paperheight);
7         \fill[color=orange] (0,0) rectangle(2.95,1.9);
8         \fill[color=blue!50!cyan!80] (3.05,0) rectangle(\the\paperwidth,1.9);
9         \ifnum\thepage>1\relax%
10            \fill[white,opacity=1] (0,0) rectangle(\the\paperwidth,\the\paperheight);
11        \fi
12    \end{tikzpicture}
13 }
14
15 % Title page
16 \defbeamertemplate*{title page}{texsx}[1][]
17 {
18     \vskip6cm%
19     \begin{beamercolorbox}[wd=12cm,leftskip=3cm,sep=8pt,#1]{title
20         page header}
21         \usebeamerfont{title}\inserttitle\par%
22     \end{beamercolorbox}%
23     \vskip0.75cm%
24     \begin{beamercolorbox}[wd=12cm,leftskip=3cm,#1]{author}
25         \usebeamerfont{author}\insertauthor%
```

```

26 \vskip0.2cm%
27 \begin{beamercolorbox}[wd=12cm,leftskip=3cm,#1]{
    date}
28 \usebeamerfont{author}\insertdate%
29 \end{beamercolorbox}
30 \vfill
31 }
32
33 \mode
34 <all>

```

Il est important de remarquer que l'on définit manuellement la boîte délimitante avec la commande `\useasboundingbox (0,0) rectangle(\the\paperwidth,\the\paperheight);`, ensuite, en utilisant un test sur le numéro de page, on peut appliquer ce style uniquement pour la page de titre.

On dispose ensuite les éléments du template title page où il faut. De plus, on caractérise les couleurs des différents éléments `beamercolorboxes`, title page header, author et date. Il nous faut donc ces définitions que l'on ajoute à `beamercolorthemetextsx.sty`.

```

1 \mode<presentation>
2
3 % Settings
4 \setbeamercolor*{title page header}{fg=white}
5 \setbeamercolor*{author}{fg=white}
6 \setbeamercolor*{date}{fg=white}
7
8 \mode
9 <all>

```

En compilant, on obtient alors une slide de titre avec l'aspect défini comme à la Figure ??.



FIGURE 1 • Aspect de la slide de titre du template

Nous pouvons alors regarder le cas des slides du corps de la présentation.

5.1.5 Slide générique

Aspect du titre des slides Ceci se fait via un template appelé `frametitle` situé dans `beamerouterthemetextsx.sty` par exemple :

```

1 \mode<presentation>
2
3 % Frame title
4 \defbeamer*{frametitle}{texsx}[1][ ]
5 {

```

```

6 \vskip1cm%
7 \begin{beamercolorbox}[wd=\paperwidth,ht=1.2cm]{frametitle}
8 \begin{tikzpicture}
9 \useasboundingbox[fill=white](0,0) rectangle(\the\paperwidth,1.2);
10 \fill[orange] (0,0) rectangle(2.95,1.2);
11 \fill[blue!50!cyan!80] (3.05,0) rectangle(\the\paperwidth,1.2);
12 \ifx\insertframesubtitle\@empty%
13 {\node[anchor=west, white,font=\large] at (3.2,0.61){\
14 insertframetitle};}
15 \else%
16 {\node[anchor= west, white,font=\large] at (3.2,0.81){\
17 insertframetitle};%
18 \node[anchor= west, white,font=\small] at (3.2,0.41){\
19 insertframesubtitle};}%
20 \fi
21 \end{tikzpicture}
22 \end{beamercolorbox}
23 }
24
25 \mode<all>

```

Pour faire simple, on exploite le même concept que pour `titlepage` pour dessiner quelques boîtes, puis on vérifie s'il y a ou non un sous-titre. On définit alors la position du titre et du sous-titre en fonction de cette réponse. Le tout est géré avec des nœuds TikZ.

Divers éléments Dans le document d'exemple proposé, on remarque la présence de listes. Ces dernières peuvent être paramétrées avec la commande :

```

1 \setbeamercolor*{item}{fg=orange}

```

Cette commande est à ajouter au fichier `beamercolorthemetexsx.sty`. On ajoute également la définition des formes des items dans le fichier `beamerinnerthemetexsx.sty` :

```

1 % Items
2 \setbeamertemplate{items}[square]
3 \setbeamertemplate{sections/subsections in toc}[square]

```

Il n'est pas obligatoire que ces déclarations soient dans ce fichier, ceci semble cependant être une norme dans la communauté qui a développé les principaux thèmes Beamer.

On peut alors compiler l'intégralité de la présentation avec notre thème. Des modifications et améliorations peuvent être apportées au thème, qui sont laissées au lecteur intéressé de le faire.