

## Présenter du code source avec LaTeX

### 9.1 Verb et verbatim

Si on veut imprimer plusieurs lignes de code source, le plus simple est d'utiliser l'environnement `{verbatim}`. Par exemple,

```
\begin{verbatim}
for i from 1 to n do
  L:=[op(L),i]
od;
\end{verbatim}
```

donnera

```
for i from 1 to n do
  L:=[op(L),i]
od;
```

La commande `\verb` permet d'afficher du code source dans le texte. C'est une commande à la syntaxe différente de celles qu'on a vu jusqu'ici. On choisit d'abord un caractère qui n'est pas utilisé dans ce qu'on veut afficher verbatim, par exemple ' , " ou | et ensuite, on tape `\verb"\{#}"` pour obtenir `\{#}`. Pour obtenir les bons caractères ' , et " et ` , utiliser le package `upquote`.

```
\usepackage{upquote}
```

Noter que `\verb` (pas plus de `{verbatim}`) ne peut pas être utilisé dans l'argument d'une commande. Il faut donc l'employer avec précaution.

**Exercice 1.** — Taper les codes suivants avec la commande `\verb`.

- `\newcommand{\abs}{\left|#1\right|}`
- `\verb"\carcode`|=13"` (le code utilisé devra avoir deux `\verb` imbriqués)

**Solution de l'exercice 1.** — Le premier s'obtient avec

```
\verb"\newcommand{\abs}{\left|#1\right|}"
```

et le second avec, par exemple

```
\verb!\verb"\carcode\`|=13"!
```

On peut bien sûr remplacer ! par, disons, & si on veut.

## 9.2 Package listings

Le problème avec `{verbatim}` est qu'il ne permet pas de colorer syntaxiquement le code source. Par exemple, le code

```
\titlecontents{section}%
  [3.8em]% retrait gauche
  {\addvspace{0pt}}% materiel avant
  {\contentslabel{2.3em}}% avant lorsqu'il y a un numero
  {\hspace{-2.3em}}% avant lorsqu'il n'y a pas de numero
  {\titlerule*[0.75em]{.}\contentspage}% pts suspension et no de page
  [\addvspace{0pt}}% materiel apres
```

n'est pas aussi lisible qu'il pourrait l'être, comme dans

```
\titlecontents{section}%
  [3.8em]% retrait gauche
  {\addvspace{0pt}}% materiel avant
  {\contentslabel{2.3em}}% avant lorsqu'il y a un numero
  {\hspace{-2.3em}}% avant lorsqu'il n'y a pas de numero
  {\titlerule*[0.75em]{.}\contentspage}% pts suspension et no de page
  [\addvspace{0pt}}% materiel apres
```

Attention, il ne faut pas mettre de lettres accentuées à l'intérieur d'un `{lstlisting}` (c'est une limitation du package listings lorsqu'on utilise l'encodage UTF8).

Le package listings propose la commande `\lstinline` (qui fonctionne comme `\verb`), l'environnement `{lstlisting}` (qui fonctionne comme `{verbatim}`) et une commande `\lstinputlisting` qui permet d'inclure un fichier externe

```
\lstinputlisting[language=HTML]{index.html}
```

Comme le montre l'exemple précédent, on peut mettre un certain nombre d'options entre crochets qui permettent de personnaliser la présentation ainsi que d'adapter le comportement du package listings à la nature du code que l'on tape.

Comme on va le voir, le package listings est relativement puissant, mais il ses réglages par défaut sont malheureusement extrêmement contestables. Par exemple, le code précédent est

```
\titlecontents{section}%
  [3.8em]% retrait gauche
  {\addvspace{0pt}}% materiel avant
  {\contentslabel{2.3em}}% avant lorsqu'il y a un numero
  {\hspace{-2.3em}}% avant lorsqu'il n'y a pas de numero
  {\titlerule*[0.75em]{.}\contentspage}% pts suspension et no de page
  [\addvspace{0pt}}% materiel apres
```

En plus de l'aspect esthétique inacceptable (police de caractère proportionnelle forcée à être monospace), la copie d'un bout de code à partir du PDF est impossible (par exemple le mot `\addvspace` précédent sort comme `\ addvspa c e`) et le grand espacement qu'il y a autour des apostrophes ou des virgules peut conduire à une ambiguïté sur la présence ou non d'une espace avant ou après ces symboles. Pour régler tous ces problèmes, on veillera, dans le préambule, après avoir chargé le package avec

```
\usepackage{listings}
```

à toujours mettre les lignes suivantes :

```
\lstset{
  upquote=true,
  columns=flexible,
  basicstyle=\ttfamily,
}
```

Le code précédent ressemblera alors à

```
\titlecontents{section}%
  [3.8em]% retrait gauche
  {\advspace{0pt}}% materiel avant
  {\contentslabel{2.3em}}% avant lorsqu'il y a un numero
  {\hspace{-2.3em}}% avant lorsqu'il n'y a pas de numero
  {\titledrule*[0.75em]{.}\contentspage}% pts suspension et no de page
  [\advspace{0pt}}% materiel apres
```

On est revenu à la même chose que l'exemple verbatim ci-dessus. Pour obtenir la coloration syntaxique, il faut spécifier le langage utilisé ainsi que les styles adaptés pour les mots-clefs. Pour fixer ces paramètres pour tout le document, il suffit de rajouter au préambule

```
\lstset{
  language=[LaTeX]TeX,
  texcsstyle=*\color{blue},
  commentstyle=\color{gray},
}
```

ce qui donnera

```
\titlecontents{section}%
  [3.8em]% retrait gauche
  {\advspace{0pt}}% materiel avant
  {\contentslabel{2.3em}}% avant lorsqu'il y a un numero
  {\hspace{-2.3em}}% avant lorsqu'il n'y a pas de numero
  {\titledrule*[0.75em]{.}\contentspage}% pts suspension et no de page
  [\advspace{0pt}}% materiel apres
```

On constate qu'il manque des mots-clefs, à savoir ceux définis par le package `\titletoc`. Pour rectifier ceci, il suffit de rajouter les mots-clefs manquant en faisant :

```
\lstset{
  moretexcs={titlecontents,contentslabel,titledrule,contentspage}
}
```

ce qui donnera ce qu'on veut, à savoir

```
\titlecontents{section}%
  [3.8em]% retrait gauche
  {\advspace{0pt}}% materiel avant
  {\contentslabel{2.3em}}% avant lorsqu'il y a un numero
  {\hspace{-2.3em}}% avant lorsqu'il n'y a pas de numero
  {\titledrule*[0.75em]{.}\contentspage}% pts suspension et no de page
  [\advspace{0pt}}% materiel apres
```

Si on souhaite que les changement ne concernent pas tous les environnements `{\lstlisting}` et tous les `\lstinline`, on peut les mettre dans un argument l'optionnel de ces environnements, en faisant attention de mettre aux éventuels [ et ] qui traînent. Par exemple, on écrira

```
\begin{lstlisting}[language={[LaTeX]TeX}]
...
\end{lstlisting}
```

et non

```
\begin{lstlisting}[language=[LaTeX]TeX]
...
\end{lstlisting}
```

(Dans ce cas, `{lstlisting}` croît que son argument optionnel est `[language=[LaTeX]TeX]` car il arrête sa lecture au premier crochet fermant.)

On peut aussi définir des environnements et commandes spéciales qui ont chacun leur réglage, ce qui est utile si on utilise fréquemment deux langages de programmation. À titre d'exemple,

```
\lstnewenvironment{latexcode}[1][\]{\lstset{language=[LaTeX]TeX,#1}}{}
\newcommand{\ltxcode}[1][\]{\lstinline[language=[LaTeX]TeX,#1]}
```

On peut bien sûr mettre autant d'options qu'on veut. Ceci permettra d'utiliser `\ltxcode"#&^_"` à la place de `\lstinline"#&^_"` tout en gardant la possibilité de spécifier des options supplémentaires en argument optionnel et de même pour

```
\begin{latexcode}
#&^_
\end{latexcode}
```

**Exercice 2.** — Définir un environnement `{code-latex}` tel que

```
\begin{code-latex}
\usepackage[runin]{abstract}
\abslabeldelim{.~---}
\setlength{\abstitlekip}{-\parindent}
%...
\begin{abstract}
Bla bla bla...
\end{abstract}
\end{code-latex}
```

affiche

```
\usepackage[runin]{abstract}
\abslabeldelim{.~---}
\setlength{\abstitlekip}{-\parindent}
%...
\begin{abstract}
Bla bla bla...
\end{abstract}
```

**Solution de l'exercice 2.** — Voici le code utilisé :

```
\documentclass{article}%          autres choix : report, book

\usepackage[utf8]{inputenc}%      encodage du fichier source
\usepackage[T1]{fontenc}%         gestion des accents (pour les pdf)
```

```

\usepackage[français]{babel}%      autres : english, greek, etc.
\usepackage{textcomp}%             caracteres additionnels
\usepackage{amsmath,amssymb}%      pour les maths
\usepackage{lmodern}%              variantes : txfonts, fourier, etc.
\usepackage[a4paper]{geometry}%     taille correcte du papier
\usepackage{graphicx}%             pour inclure des images
\usepackage{xcolor}%               pour gerer les couleurs
\usepackage{microtype}%            ameliorations typographiques

\usepackage{listings}%             pour le code informatique

\usepackage{hyperref}%             gestion des hyperliens
\hypersetup{pdfstartview=XYZ}%     zoom par default

\lstnewenvironment{code-latex}[1][]{
  \lstset{
    upquote=true,
    columns=flexible,
    basicstyle=\ttfamily,
    language={ [LaTeX]TeX},
    texcsstyle=*\color{blue},
    commentstyle=\color{gray},
    moretexcs={abslabeldelim,setlength,abstitlekip},
    #1
  }
}{}

\begin{document}

\begin{code-latex}
\usepackage[runin]{abstract}
\abslabeldelim{.~---}
\setlength{\abstitlekip}{-\parindent}
%...
\begin{abstract}
Bla bla bla...
\end{abstract}
\end{code-latex}

\end{document}

```

Voici une petite liste des langages les plus courants qui sont disponibles :

[LaTeX]TeX	HTML	R	Python	Caml	Basic
Metapost	XML	Scilab	Ruby	[Objective]Caml	Cobol
Postscript	PHP	MuPad	Perl	Java	Delphi
PSTricks	SQL	Matlab	Lisp	C	Fortran
command.com		Mathematica	VBScript	C++	Pascal
bash		Gnuplot		Assembler	

(Le langage `command.com` est le batch de Windows.)

Pour modifier l'espace autour d'un environnement `\lstlisting`, on utilise les paramètres suivants :

```
\lstset{
  xleftmargin=2em, % espace a gauche
  xrightmargin=2em, % espace a droite
  aboveskip=\topsep, % espace au-dessus
  belowskip=\topsep, % espace en-dessous
}
```

Pour modifier la présentation, on dispose de divers options pour encadrer ou mettre sur fond coloré les environnements `{\lstlisting}`. Par exemple, dans ce document, on a utilisé

```
\lstset{
  frame=single,
  rulecolor=\color{green!5},
  backgroundcolor=\color{green!5},
}
```

Les autres options de présentation sont dans la documentation.

Il est possible d'activer la coupure automatique des lignes trop longues ainsi que l'indentation à leur appliquer (compter 0.5em par caractère)

```
\lstset{
  breaklines,
  breakindent=1.5em, % indente de 3 caracteres vers la droite
}
```

Il est également possible de faire une action avant et après chaque coupure avec les clefs `postbreak` et `prebreak`.

Dans le langage TeX, les mots-clefs sont contrôlés par `texcsstyle` et `moretexcs`, mais dans la plupart des autres langages, c'est `keywordstyle` et `morekeywords` qu'il faut utiliser. Il est également possible de montrer les espaces par des «`␣`» en utilisant `showspaces=true`.

**Exercice 3.** — Sachant que le code suivant est en HTML, obtenir la présentation suivante

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD_HTML_4.01_Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>

  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-
  type">
  <title>Titre de la page web</title>
  <style type="text/css" media="all">@import "css/style.css";</style>

</head>
<!-- debut du document -->
<body>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
  eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
  ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
  aliquip ex ea commodo consequat. Duis aute irure dolor in
```


```

reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
culpa qui officia deserunt mollit anim id est laborum.</p>

<center>Source : <a href="http://www.lipsum.com/">lipsum.com</a></
center>

</body>
</html>

```

Le fichier source est attaché ci-contre. 

**Solution de l'exercice 3.** — Voici les paramètres utilisés, à mettre dans un `\lstset` ou dans un environnement défini exprès :

```

upquote=true,
columns=flexible,
basicstyle=\ttfamily,
language=HTML,
keywordstyle=\color{blue},
commentstyle=\color{gray},
breaklines,
breakindent=1.5em,
xleftmargin=2em,
xrightmargin=2em,
frame=single,
rulecolor=\color{orange},
backgroundcolor=\color{orange!5},

```

Il est possible de faire plein d'autres choses avec `listings`, comme numéroter les lignes, définir plusieurs types de mots-clés colorés différemment, définir des styles, des langages non disponibles, etc. On renvoie à la documentation.

### 9.3 Compléments

Un des défauts de `listings` est que si on indente proprement son code comme par exemple

```

\titlecontents{section}%
  [3.8em]% retrait gauche
  {\addvspace{0pt}}% materiel avant
  {\contentslabel{2.3em}}% avant lorsqu'il y a un numero
  {\hspace{-2.3em}}% avant lorsqu'il n'y a pas de numero
  {\titlerule*[0.75em]{.}\contentspage}% pts suspension et no de page
  [\addvspace{0pt}}% materiel apres

```

les espaces au début des lignes 2 à 7 sont perdus lors d'un copié collé. Pour éviter cela, on peut mettre les quelques lignes suivantes dans le préambule (cela revient à rendre visibles les espaces puis à les mettre de la bonne couleur pour qu'ils ne se voient pas ; si la page n'est pas blanche, il faut remplacer `white` par la couleur adéquate) :

```
\makeatletter
```

```
\def\lst@outputspace{{\ifx\lst@bkgcolor\empty\color{white}\else  
\lst@bkgcolor\fi\lst@visiblespace}}  
\lst@keepspacetrue  
\makeatother
```

Ce code peut nécessiter le package `cmap` pour fonctionner correctement.