

Typesetting ‘lettrines’ in L^AT_EX 2_ε documents

Daniel FLIPO

Daniel.Flipo@univ-lille1.fr

1 Introduction

The file `lettrine.dtx`¹, provides a command `\lettrine` which requires two mandatory arguments, and an optional one.

Adding `\usepackage{lettrine}` in the preamble of a document defines the command `\lettrine`, the simplest use of which is `\lettrine{<letter>}{<text>}`. It produces a dropped capital `<letter>` (2 lines high), followed by `<text>` typeset in small caps, and the rest of the paragraph is wrapped around the dropped capital.

Various parameters are provided to control the size and layout of the dropped capital and match the requirements described in the books

- “Lexique des règles typographiques en usage à l’Imprimerie nationale” troisième édition (1994), ISBN-2-11-081075-0,
- “Mise en page et impression” Yves PERROUSSEAUX, ISBN-2-911220-01-3.

The parameters can be set using David Carlisle’s `keyval.sty` syntax:

- `lines=<integer>` sets how many lines the dropped capital will occupy (default=2);
- `lhang=<decimal>` ($0 \leq \text{lhang} \leq 1$) sets how much of the dropped capital’s width should hang into the margin (default=0);
- `loversize=<decimal>` ($-1 < \text{loversize} \leq 1$) enlarges the dropped capital’s height: with `loversize=0.1` its height is enlarged by 10% so that it raises above the top paragraph’s line (default=0);
- `lraise=<decimal>` does not affect the dropped capital’s height, but moves it up (if positive), down (if negative); useful with capitals like J or Q which have a positive depth, (default=0);
- `findent=<dimen>` (positive or negative) controls the horizontal gap between the dropped capital and the indented block of text (default=0pt);
- `nindent=<dimen>` shifts all indented lines, starting from the second one, horizontally by `<dimen>` (this shift is relative to the first line, default=0.5em);
- `slope=<dimen>` can be used with dropped capitals like A or V to add `<dimen>` (positive or negative) to the indentation of each line starting from the third one (no effect if `lines=2`, default=0pt);
- `ante=<text>` can be used to typeset `<text>` before the dropped capital (typical use is for French guillemets starting the paragraph).

¹The file described in this section has version number v1.62 and was last revised on 2007/08/08.

- `image=<true>` (new to version 1.6) will force `\lettrine` to replace the letter normally used as dropped capital by an image in eps format (latex) or in pdf, jpg, etc. format (pdflatex); this needs the `graphicx` package to be loaded in the preamble of course. `\lettrine[image=true]{A}{n exemple}` or just `\lettrine[image]{A}{n exemple}` will load `A.eps` or `A.pdf` instead of letter A. This was suggested by Bill Jetzer. Redefining `\LettrineFont` as `\LettrineFontEPS` still works for compatibility but is deprecated.

Example: `\lettrine[lines=4, lraise=0.1, nindent=0em, slope=-.5em]{V}{oici} un exemple ...`

Coloured lettrines are possible in conjunction with package `color`, examples: `\lettrine{\textcolor{red}{A}}{n} exemple` or `\lettrine{\textcolor[gray]{0.5}{A}}{nother} one` see package `color` for the syntax of colour commands. Another possibility to colour lettrines globally is described below, see `\LettrineFontHook`.

Once `lettrine.sty` will be installed (run `latex` on `lettrine.ins` to extract it), compile and print `demo.tex` to see the possible usage of these parameters.

The default settings can be customized either in a config file `lettrine.cfg` (for a global usage), or on a per document basis, in the preamble of each document. The following list shows the syntax to set them and their default values:

- `\setcounter{DefaultLines}{2},`
- `\renewcommand{\DefaultLoversize}{0},`
- `\renewcommand{\DefaultLraise}{0},`
- `\renewcommand{\DefaultLhang}{0},`
- `\LettrineImagefalse,`
- `\setlength{\DefaultFindent}{0pt},`
- `\setlength{\DefaultNindent}{0.5em},`
- `\setlength{\DefaultSlope}{0pt}.`

Instead of giving optional parameters to the `\lettrine` command, it is possible, from version 1.5, to set them on a per character basis in a second config file (suggested by Pascal Kockaert): `\renewcommand{\DefaultOptionsFile}{filename}` in the preamble (or anywhere in the document) will force each call to `\lettrine` to read the file `filename`. See examples of such config files in the subdirectory `contrib`.

The dimensional parameters `findent`, `nindent` and `slope`, can be set in `filename` relative to `\LettrineWidth` if needed. `\LettrineWidth` should be used *only* in the config file `filename`, its value is non-null only if `filename` is read! The settings read from this file will be overridden by the optional arguments eventually given to `\lettrine`.

`\LettrineTextFont` sets the font used for the second argument of `\lettrine`, its default definition is `\newcommand{\LettrineTextFont}{\scshape}` (second argument in small caps, this can be changed using `\renewcommand`).

`\LettrineFont` sets the font used for the dropped capital, usually the current font in a (large) size, computed automatically from the number of lines it will fill: the font size is computed so that, a *standard* dropped capital (say X, not Å) when sitting on its baseline, gets its top aligned with the top of the following text (provided `loversize = 0` and `lines ≥ 2`). When `lines = 1`, size is computed as if `lines` was 2.

A hook `\LettrineFontHook` is provided to change the font used for the dropped capital, syntax follows L^AT_EX's low-level font interface (see L^AT_EX Companion, p.187–192), the `\selectfont` command is issued by `\LettrineFont`:

```
\renewcommand{\LettrineFontHook}{\fontfamily{ppl}\fontseries{bx}}%
\fontshape{sl}},
```

selects Palatino bold expanded slanted for the dropped capital.

`\LettrineFontHook` can also be used to change the colour of all lettrines in a (part of) document: `\renewcommand{\LettrineFontHook}{\color[gray]{0.5}}` will colour the lettrines following this command in grey.

Important notice: the sizing works fine with *fully scalable* fonts (like the standard PostScript fonts), but might not work well with CM/EC fonts which have two limitations: only a limited number of sizes is available by default (precise adjustments are impossible), and the largest size (25pt or 35pt) is often too small. The CM fonts are now available in PostScript type1 format for free (courtesy of BlueSky/Y&Y), to make them fully scalable, it is mandatory to add `\usepackage{type1cm}` in the preamble of your document. The EC fonts are also available in type1 format for free (thanks to Vladimir Volovich, they are called cm-super), and adding `\usepackage{type1ec}`² in the preamble will make them fully scalable too. So, if you want `lettrine.sty` to work properly with CM or EC fonts, you will need *PostScript versions* of these fonts *and* one of the packages `type1cm.sty` or `type1ec.sty`.

The newly released LM fonts, which should replace both CM and EC fonts sooner or later, are fully scalable, so, if you have them installed, you should use them instead of CM or EC fonts. `\usepackage{lmodern}` is the command to switch them on (add `\usepackage[T1]{fontenc}` when composing in one of the western languages other than English in order to get proper hyphenation).

You can also consider using one of the standard PostScript fonts (Palatino, Times, Utopia...), they are fully scalable too!

Known problems:

- nothing is done to prevent page-breaking in a paragraph starting with a dropped capital; when it happens to hang into the footer, page-breaking has to be done manually;
- `\lettrine` works within ‘quote’ ‘quotation’, ‘abstract’ environments but does not work within ‘center’ environments (except with option `[lines=1]`);
- `\lettrine` does not work within lists;
- if a *list* has to be included in a paragraph starting with a ‘lettrine’, it is necessary to add the command `\parshape=0` just after the end of the list

²This package, available on CTAN, was first released on 2002/07/30.

(starting a new paragraph just before or just after the list works too). Remember that ‘quote’, ‘quotation’, ‘abstract’ environments are implemented as *lists* in L^AT_EX.

2 T_EXnical details

This package only runs with L^AT_EX 2_ε and requires keyval.sty

```
1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \RequirePackage{keyval}
```

Default initializations: define the necessary counters, lengths, and commands to hold the default settings and set these default settings. They can be overwritten in file `lettrine.cfg`.

```
3 \newcounter{DefaultLines}
4 \setcounter{DefaultLines}{2}
5 \newcommand*\DefaultOptionsFile{\relax}
6 \newcommand*\DefaultLoversize{0}
7 \newcommand*\DefaultLraise{0}
8 \newcommand*\DefaultLhang{0}
9 \newdimen\DefaultFindent
10 \setlength{\DefaultFindent}{\z@}
11 \newdimen\DefaultNindent
12 \setlength{\DefaultNindent}{0.5em}
13 \newdimen\DefaultSlope
14 \setlength{\DefaultSlope}{\z@}
15 \newif\ifLettrineImage
```

Then let’s define the necessary internal counters, lengths, and commands.

```
16 \newsavebox{\L@lbox}
17 \newsavebox{\L@tbox}
18 \newcounter{L@lines}
19 \newdimen\L@Pindent
20 \newdimen\L@Findent
21 \newdimen\L@Nindent
22 \newdimen\L@lraise
23 \newdimen\L@first
24 \newdimen\L@next
25 \newdimen\L@slope
26 \newdimen\L@height
27 \newcommand*\L@file{}
28 \newcommand*\L@hang{}
29 \newcommand*\L@oversize{}
30 \newcommand*\L@raise{}
31 \newcommand*\L@ante{}
32 \newif\ifL@image
```

Provide commands for the fonts used to typeset the two mandatory arguments of `\lettrine`.

`\LettrineTextFont` In French, small caps usually follow the dropped capital.

```
33 \newcommand*{\LettrineTextFont}{\scshape}
```

`\LettrineFont` The default size for the dropped capital is computed so that the top of it is exactly aligned with the top of the following text; an extra height (positive or negative) may be added with `Defaultloversize` or an optional argument `loversize=`.
`\LettrineFontHook` If `lines=1`, the default size for the dropped capital is computed as if `lines=2`.
`\LettrineFontEPS` `\Lettrine@height` computes the wished height for the dropped capital and stores it into `\L@height`. As `\baselineskip` might be a rubber length, we convert it into a ‘dimen’ using `\@tempdima`. `\LettrineFontHook` enables to select another font for the dropped capital. Its default definition is empty (the current text font is used).

```
34 \def\Lettrine@height{%
35   \@tempdima=\baselineskip
36   \setlength{\L@height}{\theL@lines\@tempdima}%
37   \ifnum\theL@lines>1
38     \addtolength{\L@height}{-\@tempdima}%
39   \fi
40   \sbox{\L@tbox}{\LettrineTextFont x}%
41   \addtolength{\L@height}{\ht\L@tbox}%
42   \addtolength{\L@height}{\L@loversize\L@height}}
43 \newcommand*{\LettrineFontHook}{}
44 \newcommand*{\LettrineFont}{%
45   \Lettrine@height
```

`\L@height` now holds the exact height required for the dropped capital, setting `\fontsize` to that height would not give the expected result (capital too small), some computing has to be done: we measure how high an ‘X’ capital would be and compute a scaling factor (always ≥ 1).

```
46   \sbox{\L@tbox}{\LettrineFontHook\fontsize{\L@height}{\L@height}%
47     \selectfont X}%
```

Arithmetic calculations convert the dimensions into integers (in sp) and compute a (4 decimal accurate) scaling factor.

```
48   \@tempcntb=\ht\L@tbox
49   \@tempcnta=\L@height
50   \multiply\@tempcnta by 100
51   \divide\@tempcntb by 100
52   \divide\@tempcnta by \@tempcntb
53   \advance\@tempcnta by -9999
54   \ifnum\@tempcnta>0
55     \def\@tempa{1.\the\@tempcnta}%
56   \else
57     \def\@tempa{1}%
58   \fi
59   \LettrineFontHook
60   \fontsize{\@tempa\L@height}{\@tempa\L@height}%
61   \selectfont
62 }
```

The following definition is for use with dropped capitals defined as images: EPS, PDF, JPG, PNG files (see examples in `demo.tex`). Its use requires the `graphicx`

package to be loaded in the preamble with `\usepackage{graphicx}`. The required size is computed just as in the standard case, `\includegraphics` prints the EPS file at this size.

```
63 \newcommand*\LettrineFontEPS}{%
64   \Lettrine@height\LettrineFontHook
65   \includegraphics[height=\L@height]}
```

Set up keyval initializations.

```
66 \define@key{L}{lines}{\setcounter{L@lines}{#1}}
67 \define@key{L}{lhang}{\renewcommand*\L@hang{#1}}
68 \define@key{L}{loversize}{\renewcommand*\L@oversize{#1}}
69 \define@key{L}{lraise}{\renewcommand*\L@raise{#1}}
70 \define@key{L}{ante}{\renewcommand*\L@ante{#1}}
71 \define@key{L}{findent}{\setlength{L@Findent}{#1}}
72 \define@key{L}{nindent}{\setlength{L@Nindent}{#1}}
73 \define@key{L}{slope}{\setlength{L@slope}{#1}}
74 \define@key{L}{image}[true]{\csname L@image#1\endcsname}
```

`\LettrineOptionsFor` This command is to be used in an optional config file (the name of which is found in `\DefaultOptionsFile`) to set the values of parameters on a per character basis, for instance:

`\LettrineOptionsFor{A}{slope=0.6em, findent=-1em, nindent=0.6em}` creates an internal command (`\l@A`-keys in this example), which expands to the options given as second argument of `\LettrineOptionsFor` for letter ‘A’.

```
75 \newcommand*\LettrineOptionsFor}[2]{\@namedef{l@#1-keys}{#2}}
76 \newdimen\LettrineWidth
```

`\lettrine` Now let’s define `\lettrine`.

```
77 \def\lettrine{\@ifnextchar[\@lettrine{\@lettrine[]}}
78 \def\@lettrine[#1]#2#3{%
```

First reset the parameters to their default values:

```
79   \setcounter{L@lines}{\theDefaultLines}%
80   \renewcommand*\L@hang{\DefaultLhang}%
81   \renewcommand*\L@oversize{\DefaultLoversize}%
82   \renewcommand*\L@raise{\DefaultLraise}%
83   \renewcommand*\L@ante{}%
84   \setlength{L@Findent}{\DefaultFindent}%
85   \setlength{L@Nindent}{\DefaultNindent}%
86   \setlength{L@slope}{\DefaultSlope}%
87   \ifLettrineImage\L@imagertrue\else\L@imagefalse\fi
```

then try to read an optional file, the name of which is found in `\DefaultOptionsFile`, do this inside a group, and define a global command `\l@LOKeys` which will expand to the list of options given by `\LettrineOptionsFor` for the current lettrine (defined by #2)...

```
88   \if\DefaultOptionsFile\relax
89   \else
90     \begingroup
```

```

91 \InputIfFileExists{\DefaultOptionsFile}%
92 {}%
93 {\PackageWarning{letrrine.sty}%
94 {File \DefaultOptionsFile\space not found}%
95 }%

```

Gobble the colour commands, just keep the letter argument.

```

96 \def\color##1##{\l@color{##1}}%
97 \let\l@color\@gobbletwo
98 \def\textcolor##1##{\l@textcolor{##1}}%
99 \def\l@textcolor##1##2##3{##3}%

```

Save the list of options relevant to the letter in #2 in \l@LOKeys (list is empty eventually).

```

100 \expandafter\ifx\csname l@#2-keys\endcsname\relax
101 \gdef\l@LOKeys{}%
102 \else
103 \xdef\l@LOKeys{\csname l@#2-keys\endcsname}%
104 \fi
105 \endgroup

```

Now apply these options (the following code executes \setkeys{L}{\l@LOKeys} where \l@LOKeys is expanded, see keyval.sty).

```

106 \def\KV@prefix{KV@L@}%
107 \let\@tempc\relax
108 \expandafter\KV@do\l@LOKeys,\relax,

```

As some parameters' values `findent`, `nindent` and `slope` — which do not influence the letrrine size — may be given relative to `\LettrrineWidth`, this has to be done again after measuring the letrrine for `\LettrrineWidth` to be set properly.

```

109 \sbox{\L@lbox}{\LettrrineFont #2}%
110 \settowidth{\LettrrineWidth}{\usebox{\L@lbox}}%
111 \def\KV@prefix{KV@L@}%
112 \let\@tempc\relax
113 \expandafter\KV@do\l@LOKeys,\relax,
114 \fi

```

Finally read the optional argument: the `keyval` package will set the parameters mentioned calling `\define@key` (see above).

```

115 \setkeys{L}{#1}%

```

Save the two mandatory arguments of `\letrrine` into two boxes, this will help measuring their sizes. Depending on the boolean `image`, `\LettrrineFont` or `\LettrrineFontEPS` is used.

```

116 \ifL@image
117 \sbox{\L@lbox}{\LettrrineFontEPS{#2}}%
118 \else
119 \sbox{\L@lbox}{\LettrrineFont #2}%
120 \fi
121 \sbox{\L@tbox}{\LettrrineTextFont{#3}}%

```

Start a new paragraph, skipping the necessary amount of space if the dropped capital sticks out the top of paragraph. We use `\L@first` to compute the amount of space to be skipped. Again, as `\baselineskip` might be a rubber length, we convert it into a ‘dimen’ using `\@tempdima`.

```

122  \@tempdima=\baselineskip
123  \ifnum\theL@lines=1
124    \setlength{\L@first}{\ht\L@lbox}%
125    \addtolength{\L@first}{-\ht\L@tbox}%
126    \setlength{\L@lraise}{\z@}%
127  \else
128    \setlength{\L@first}{-\theL@lines\@tempdima}%
129    \addtolength{\L@first}{\@tempdima}%
130    \sbox{\@tempboxa}{\LettrineTextFont x}%
131    \addtolength{\L@first}{-\ht\@tempboxa}%

```

Now, `\L@first` holds (the opposite of) the raw height of a standard dropped capital (like ‘X’), excluding the effect of `\L@oversize`. This is the basis for `\L@raise` (and `\L@oversize`, see `\LettrineFont`).

```

132  \setlength{\L@lraise}{-\L@raise\L@first}%
133  \addtolength{\L@first}{\L@lraise}%
134  \addtolength{\L@first}{\ht\L@lbox}%
135  \addtolength{\L@lraise}{-\theL@lines\@tempdima}%
136  \addtolength{\L@lraise}{\@tempdima}%
137  \fi
138  \par
139  \ifdim\L@first>0.2\p@\vskip\L@first\fi

```

Again, we (mis)use the length `\L@first` to compute the width of the text eventually coming before the dropped capital. It is reset later on to hold the first line’s length.

```

140  \setlength{\L@Pindent}{\wd\L@lbox}%
141  \addtolength{\L@Pindent}{-\L@hang\wd\L@lbox}%
142  \settowidth{\L@first}{\L@ante}%
143  \addtolength{\L@Pindent}{\L@first}%
144  \addtolength{\L@Pindent}{\L@Findent}%
145  \setlength{\L@first}{\linewidth}%
146  \addtolength{\L@first}{-\L@Pindent}%

```

Now let’s compute `\L@Nindent` and `\L@next` for the next lines.

```

147  \addtolength{\L@Nindent}{\L@Pindent}%
148  \setlength{\L@next}{\linewidth}%
149  \addtolength{\L@next}{-\L@Nindent}%

```

This is for quotation, quote, abstract... environments: `\linewidth` is set by these environments, all we have to do is to shift our text left by `\rightmargin` (amount of space locally added to `\leftmargin` in these environments).

```

150  \addtolength{\L@Pindent}{\rightmargin}%
151  \addtolength{\L@Nindent}{\rightmargin}%

```

Now, set up the shape of the new paragraph (designed by `\parshape`).

```

152  \addtocounter{L@lines}{1}%

```



```

153 \def\L@parshape{\c@L@lines \the\L@Pindent \the\L@first}%
154 \@tempcnta=\tw@
155 \@whilenum \@tempcnta<\c@L@lines\do{%
156   \edef\L@parshape{\L@parshape \the\L@Nindent \the\L@next}%
157   \addtolength{\L@Nindent}{\L@slope}%
158   \addtolength{\L@next}{-\L@slope}%
159   \advance\@tempcnta\@ne}%
160 \edef\L@parshape{\L@parshape \rightmargin \the\linewidth}%
161 \noindent\leavevmode
162 \parshape=\L@parshape

```

Write the dropped capital into the left margin, and wrap the rest of paragraph around it.

```

163 $\smash{\llap{\mbox{\L@ante}\raisebox{\L@lraise}{\usebox{\L@lbox}}}%
164   \hskip \the\L@Findent}}$\%
165 \usebox{\L@tbox}}

```

This ends the definition of `\lettrine`.

Load a local config file if present in L^AT_EX's search path.

```

166 \InputIfFileExists{lettrine.cfg}
167   {\typeout{Loading lettrine.cfg}}
168   {\typeout{lettrine.cfg not found, using default values}}

```

3 Fichier de configuration

```

169 %% lettrine.cfg: configuration file for lettrine.sty
170 %%
171 %% If you want to customize lettrine, please do not hack into the
172 %% code, copy this file to the directory where lettrine.sty lies
173 %% and customize it as you like.
174 %%
175 %% Uncomment these lines and change the parameters' values to fit
176 %% your needs (see lettrine.dtx).
177 %%
178 %%\setcounter{DefaultLines}{2}
179 %%
180 %% These are decimal numbers:
181 %%\renewcommand{\DefaultLoversize}{0}
182 %%\renewcommand{\DefaultLraise}{0}
183 %%\renewcommand{\DefaultLhang}{0}
184 %%
185 %% These are lengths (don't forget the unit):
186 %%\setlength{\DefaultFindent}{0pt}
187 %%\setlength{\DefaultNindent}{0.5em}
188 %%\setlength{\DefaultSlope}{0mm}
189 %%
190 %% This is a flag (value=true/false):
191 %%\LettrineImagefalse
192 %%
193 %% In case you want to set parameters for some letters
194 %% in file 'optfile.cfl'
195 %%\renewcommand{\DefaultOptionsFile}{optfile.cfl}

```

Change History

lettrine-0.81		fonts and the typelec package by Vladimir Volovich.	3
<code>\lettrine: \DefaultLoversize</code> added.	6		
lettrine-0.9		lettrine-1.4	
<code>\lettrine: Calculations of length</code> <code>\L@first</code> changed. Do not ‘vskip’ small lengths (<0.2pt), they are just rounding errors.	8	<code>\lettrine: \lettrine</code> still didn’t work properly in quote, quo- tation, abstract environments, pointed out by Matthias C. Schmidt. <code>\rightmargin</code> was added too early to <code>\L@Nindent</code> , thus making <code>\L@next</code> too short by <code>\rightmargin</code>	8
<code>\LettrineFontEPS: \Lettrine@height</code> added.	5	lettrine-1.5	
<code>\LettrineFontEPS</code> added.	5	General: <code>\LettrineOptionsFor</code> and <code>\LettrineWidth</code> added.	6
<code>\LettrineFontHook</code> added.	5	<code>\lettrine: Added reading</code> of an optional config file <code>\DefaultOptionsFile</code>	6
1 changed (was <code>\Huge</code>).	5	lettrine-1.6	
lettrine-1.1		General: Add a flag to switch to im- ages in eps or pdf format. Sug- gested by Bill Jetzer.	1
<code>\lettrine: Add \rightmargin</code> to <code>\L@Pindent</code> for <code>\Lettrine</code> to work properly in quote, quota- tion, abstract environments... but do not change <code>\linewidth</code> which is set by these environ- ments.	8	Added newif <code>\ifL@image</code>	4
lettrine-1.2		Added newif <code>\ifLettrineImage</code>	4
General: <code>\newlength</code> changed to <code>\newdimen</code> , to correct a bug with seminar.cls (pointed out by Peter Münster).	4	<code>\lettrine: Add braces around</code> <code>#3</code> to allow commands taking an argument (such as <code>\MakeLowercase</code>) in <code>\LettrineTextFont</code> . Suggested by Philipp Lehman.	7
<code>\lettrine: \baselineskip</code> may be a rubber length, we convert it to a dimen.	8	<code>\LettrineFontEPS: Added</code> newif <code>\LettrineFontHook</code> to <code>\LettrineFontEPS</code> . This is needed for color options.	6
<code>\LettrineFontEPS: \baselineskip</code> may be a rubber length, we con- vert it to a dimen.	5		
lettrine-1.3			
General: Correct the documenta- tion to mention the cm-super			