Typesetting 'lettrines' in LaTeX 2ε 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>` ($0 \leq$ `lhang` ≤ 1) sets how much of the dropped capital's width should hang into the margin (default=0);

- `loversize=<decimal>` ($-1 <$ `loversize` ≤ 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);

---

[1] The file described in this section has version number v1.6 and was last revised on 2004/05/22.

- `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).

- `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 conjonction with package `color`, examples:
`\lettrine{\textcolor{red}{A}}{n} example`   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 À) 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 LaTeX's low-level font interface (see LaTeX 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

---

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

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

2   TeXnical details

This package only runs with LaTeX 2$_\varepsilon$ and requires keyval.sty

```
1 ⟨∗code⟩
2 %% Please report errors to: Daniel.Flipo@univ-lille1.fr
3 %%
4 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
5 \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`.

```
6 \newcounter{DefaultLines}
7 \setcounter{DefaultLines}{2}
8 \newcommand*{\DefaultOptionsFile}{\relax}
9 \newcommand*{\DefaultLoversize}{0}
10 \newcommand*{\DefaultLraise}{0}
11 \newcommand*{\DefaultLhang}{0}
```

```
12 \newdimen\DefaultFindent
13 \setlength{\DefaultFindent}{\z@}
14 \newdimen\DefaultNindent
15 \setlength{\DefaultNindent}{0.5em}
16 \newdimen\DefaultSlope
17 \setlength{\DefaultSlope}{\z@}
18 \newif\ifLettrineImage
```

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

```
19 \newsavebox{\L@lbox}
20 \newsavebox{\L@tbox}
21 \newcounter{L@lines}
22 \newdimen\L@Pindent
23 \newdimen\L@Findent
24 \newdimen\L@Nindent
25 \newdimen\L@lraise
26 \newdimen\L@first
27 \newdimen\L@next
28 \newdimen\L@slope
29 \newdimen\L@height
30 \newcommand*{\L@file}{}
31 \newcommand*{\L@hang}{}
32 \newcommand*{\L@oversize}{}
33 \newcommand*{\L@raise}{}
34 \newcommand*{\L@ante}{}
35 \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.

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

\LettrineFont
\LettrineFontHook
\LettrineFontEPS
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=`. If `lines=1`, the default size for the dropped capital is computed as if `lines=2`. `\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).

```
37 \def\Lettrine@height{%
38    \@tempdima=\baselineskip
39    \setlength{\L@height}{\theL@lines\@tempdima}%
40    \ifnum\theL@lines>1
41       \addtolength{\L@height}{-\@tempdima}%
```

```
42    \fi
43    \sbox{\L@tbox}{\LettrineTextFont x}%
44    \addtolength{\L@height}{\ht\L@tbox}%
45    \addtolength{\L@height}{\L@oversize\L@height}}
46 \newcommand*{\LettrineFontHook}{}
47 \newcommand*{\LettrineFont}{%
48    \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).

```
49    \sbox{\L@tbox}{\LettrineFontHook\fontsize{\L@height}{\L@height}%
50                   \selectfont X}%
```

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

```
51    \@tempcntb=\ht\L@tbox
52    \@tempcnta=\L@height
53    \multiply\@tempcnta by 100
54    \divide\@tempcntb by 100
55    \divide\@tempcnta by \@tempcntb
56    \advance\@tempcnta by -9999
57    \ifnum\@tempcnta>0
58      \def\@tempa{1.\the\@tempcnta}%
59    \else
60      \def\@tempa{1}%
61    \fi
62    \LettrineFontHook
63    \fontsize{\@tempa\L@height}{\@tempa\L@height}%
64    \selectfont
65 }
```

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.

```
66 \newcommand*{\LettrineFontEPS}{%
67    \Lettrine@height\LettrineFontHook
68    \includegraphics[height=\L@height]}
```

Set up keyval initializations.

```
69 \define@key{L}{lines}{\setcounter{L@lines}{#1}}
70 \define@key{L}{lhang}{\renewcommand*{\L@hang}{#1}}
71 \define@key{L}{loversize}{\renewcommand*{\L@oversize}{#1}}
72 \define@key{L}{lraise}{\renewcommand*{\L@raise}{#1}}
```

```
73 \define@key{L}{ante}{\renewcommand*{\L@ante}{#1}}
74 \define@key{L}{findent}{\setlength{\L@Findent}{#1}}
75 \define@key{L}{nindent}{\setlength{\L@Nindent}{#1}}
76 \define@key{L}{slope}{\setlength{\L@slope}{#1}}
77 \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'.

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

\lettrine   Now let's define \lettrine.

```
80 \def\lettrine{\@ifnextchar[\@lettrine{\@lettrine[]}}
81 \def\@lettrine[#1]#2#3{%
```

First reset the parameters to their default values:

```
82    \setcounter{L@lines}{\theDefaultLines}%
83    \renewcommand*{\L@hang}{\DefaultLhang}%
84    \renewcommand*{\L@oversize}{\DefaultLoversize}%
85    \renewcommand*{\L@raise}{\DefaultLraise}%
86    \renewcommand*{\L@ante}{}%
87    \setlength{\L@Findent}{\DefaultFindent}%
88    \setlength{\L@Nindent}{\DefaultNindent}%
89    \setlength{\L@slope}{\DefaultSlope}%
90    \ifLettrineImage\L@imagetrue\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 ex-
pand to the list of options given by \LettrineOptionsFor for the current lettrine
(defined by #2)...

```
91    \if\DefaultOptionsFile\relax
92    \else
93      \begingroup
94      \InputIfFileExists{\DefaultOptionsFile}%
95        {}%
96        {\PackageWarning{lettrine.sty}%
97          {File \DefaultOptionsFile\space not found}%
98        }%
```

Gobble the colour commands, just keep the letter argument.

```
99      \def\color##1##{\l@color{##1}}%
100     \let\l@color\@gobbletwo
```

7

```
101      \def\textcolor##1##{\l@textcolor{##1}}%
102      \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).

```
103      \expandafter\ifx\csname l@#2-keys\endcsname\relax
104              \gdef\l@LOKeys{}%
105           \else
106              \xdef\l@LOKeys{\csname l@#2-keys\endcsname}%
107           \fi
108      \endgroup
```

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

```
109      \def\KV@prefix{KV@L@}%
110      \let\@tempc\relax
111      \expandafter\KV@do\l@LOKeys,\relax,
```

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

```
112      \sbox{\L@lbox}{\LettrineFont #2}%
113      \settowidth{\LettrineWidth}{\usebox{\L@lbox}}%
114      \def\KV@prefix{KV@L@}%
115      \let\@tempc\relax
116      \expandafter\KV@do\l@LOKeys,\relax,
117   \fi
```

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

```
118   \setkeys{L}{#1}%
```

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

```
119   \ifL@image
120      \sbox{\L@lbox}{\LettrineFontEPS{#2}}%
121   \else
122      \sbox{\L@lbox}{\LettrineFont #2}%
123   \fi
```

```
124   \sbox{\L@tbox}{\LettrineTextFont{#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`.

8

```
125    \@tempdima=\baselineskip
126    \ifnum\theL@lines=1
127      \setlength{\L@first}{\ht\L@lbox}%
128      \addtolength{\L@first}{-\ht\L@tbox}%
129      \setlength{\L@lraise}{\z@}%
130    \else
131      \setlength{\L@first}{-\theL@lines\@tempdima}%
132      \addtolength{\L@first}{\@tempdima}%
133      \sbox{\@tempboxa}{\LettrineTextFont x}%
134      \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`).

```
135      \setlength{\L@lraise}{-\L@raise\L@first}%
136      \addtolength{\L@first}{\L@lraise}%
137      \addtolength{\L@first}{\ht\L@lbox}%
138      \addtolength{\L@lraise}{-\theL@lines\@tempdima}%
139      \addtolength{\L@lraise}{\@tempdima}%
140    \fi
141    \par
142    \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.

```
143    \setlength{\L@Pindent}{\wd\L@lbox}%
144    \addtolength{\L@Pindent}{-\L@hang\wd\L@lbox}%
145    \settowidth{\L@first}{\L@ante}%
146    \addtolength{\L@Pindent}{\L@first}%
147    \addtolength{\L@Pindent}{\L@Findent}%
148    \setlength{\L@first}{\linewidth}%
149    \addtolength{\L@first}{-\L@Pindent}%
```

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

```
150    \addtolength{\L@Nindent}{\L@Pindent}%
151    \setlength{\L@next}{\linewidth}%
152    \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).

```
153    \addtolength{\L@Pindent}{\rightmargin}%
154    \addtolength{\L@Nindent}{\rightmargin}%
```

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

```
155    \addtocounter{L@lines}{1}%
156    \def\L@parshape{\c@L@lines \the\L@Pindent \the\L@first}%
```

```
157  \@tempcnta=\tw@
158  \@whilenum \@tempcnta<\c@L@lines\do{%
159      \edef\L@parshape{\L@parshape \the\L@Nindent \the\L@next}%
160      \addtolength{\L@Nindent}{\L@slope}%
161      \addtolength{\L@next}{-\L@slope}%
162      \advance\@tempcnta\@ne}%
163  \edef\L@parshape{\L@parshape \rightmargin \the\linewidth}%
164  \noindent\leavevmode
165  \parshape=\L@parshape
```

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

```
166  $\smash{\llap{\mbox{\L@ante}\raisebox{\L@lraise}{\usebox{\L@lbox}}%
167          \hskip \the\L@Findent}}$%
168  \usebox{\L@tbox}}
```

This ends the definition of \lettrine.
Load a local config file if present in LaTeX's search path.

```
169  \InputIfFileExists{lettrine.cfg}{}{}
170  ⟨/code⟩
```