L'extension lettrine[*]
Composition de « lettrines »
dans des documents LaTeX 2ε

Daniel Flipo
daniel.flipo@free.fr

1 Introduction

✖The file `lettrine.dtx` 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) ;
- `depth=`<*integer*> sets the number of lines to be reserved under the baseline, this is meant for dropped capital with positive depth, like Q (default=0) ;
- `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) ;
- `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) ;

[*] Ce fichier a pour numéro de version v1.9 et a été mis à jour le 31/08/2015. Son titre original est « *Typesetting 'lettrines' in LaTeX 2ε documents* ».

- `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.
- `grid=`<*true*> (new to version 1.8) will force the vertical skip added above the paragraph starting with `\lettrine` to be rounded up to an integer number of `\baselineskip`. This option is meant for grid typesetting.
- `novskip=`<*dimen*> (new to version 1.8) overrides `\DiscardVskip` (default=0.2pt). In some cases (see options `lraise`, `loversize` or accentuated dropped capitals,...) the top of the dropped capital will raise above the top of following text (usually in small caps), this will trigger a corresponding vertical skip above the paragraph starting with `\lettrine`, *only if* this skip exceeds `\DiscardVskip`. Consider enlarging `novskip` (or `\DiscardVskip`) to prevent small vertical skips from being rounded up to `\baselineskip` when using the 'grid' option.
- `realheight=`<*true*> (new to version 1.9) will compute the default height of the lettrine so that the top of it is exactly aligned with the top of the text entered as second mandatory argument of `\lettrine` taking possible accents into account. Otherwise, the default height is computed using a customisable string `\LettrineSecondString` instead of the real argument. For backward compatibility, option `realheight` defaults to false and `\LettrineSecondString` to 'x'.

  You probably don't need this option if you choose to typeset the second mandatory argument of `\lettrine` in small caps (the default). If you change `\LettrineTextFont` to `\relax` or `\upshape`, consider these two examples :

  `\lettrine{H}{ello}` you probably would like the top of the 'L' to be aligned with the top of the 'll' rather than with the top of the 'e', adding option `realheight` does the trick : `\lettrine[realheight]{H}{ello}`. Global variants : `\LettrineRealHeighttrue` or (without the `realheight` option) `\renewcommand{\LettrineSecondString}{l}`.

  `\lettrine{L}{a misère}` option `realheight=true` would align with the top of the 'L' with the top of the grave accent, the default is probably better (top of the 'L' aligned with the top of the non accented letters).

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`.

Three dimensions, `\LettrineWidth`, `\LettrineHeight` and `\LettrineDepth`, store the final size of the lettrine.

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}`,
- `\setcounter{DefaultDepth}{0}`,
- `\renewcommand*{\DefaultLoversize}{0}`,
- `\renewcommand*{\DefaultLraise}{0}`,
- `\renewcommand*{\DefaultLhang}{0}`,
- `\LettrineImagefalse`,
- `\LettrineOnGridfalse`,
- `\LettrineRealHeightfalse`,
- `\setlength{\DefaultFindent}{0pt}`,
- `\setlength{\DefaultNindent}{0.5em}`,
- `\setlength{\DefaultSlope}{0pt}`.
- `\setlength{\DiscardVskip}{0.2pt}`.

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. 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}`[1] 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 LM fonts are a good replacement for both CM and EC fonts they are fully scalable, so 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...), or any OpenType font, 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

```
\NeedsTeXFormat{LaTeX2e}[1999/12/01]
\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`.

```
\newcounter{DefaultLines}
\setcounter{DefaultLines}{2}
\newcounter{DefaultDepth}
\newcommand*{\DefaultOptionsFile}{\relax}
\newcommand*{\DefaultLoversize}{0}
```

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

```
\newcommand*{\DefaultLraise}{0}
\newcommand*{\DefaultLhang}{0}
\newdimen\DefaultFindent
\setlength{\DefaultFindent}{\z@}
\newdimen\DefaultNindent
\setlength{\DefaultNindent}{0.5em}
\newdimen\DefaultSlope
\setlength{\DefaultSlope}{\z@}
\newdimen\DiscardVskip
\setlength{\DiscardVskip}{0.2\p@}
\newif\ifLettrineImage
\newif\ifLettrineOnGrid
\newif\ifLettrineRealHeight
```

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

```
\newsavebox{\L@lbox}
\newsavebox{\L@tbox}
\newcounter{L@lines}
\newcounter{L@depth}
\newdimen\L@Pindent
\newdimen\L@Findent
\newdimen\L@Nindent
\newdimen\L@lraise
\newdimen\L@first
\newdimen\L@next
\newdimen\L@slope
\newdimen\L@height
\newdimen\L@novskip
\newcommand*{\L@file}{}
\newcommand*{\L@hang}{}
\newcommand*{\L@oversize}{}
\newcommand*{\L@raise}{}
\newcommand*{\L@ante}{}
\newif\ifL@image
\newif\ifL@grid
\newif\ifL@realh
```

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

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

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

`\LettrineFontHook`
`\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 with 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`. `\L@height` depends only on `L@lines`, `\L@oversize` and, in case option `realheight=true` on the height of `\L@tbox`. So options *must* be read and `\L@tbox` must be properly initialised *before* executing `\Lettrine@height` (see below in `\@lettrine` code). A default initialisation of `\L@tbox` is provided just in case `\LettrineFont` would be used outside `\lettrine`.

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).

```
\def\Lettrine@height{%
   \@tempdima=\baselineskip
   \setlength{\L@height}{\value{L@lines}\@tempdima}%
   \ifnum\value{L@lines}>1
     \addtolength{\L@height}{-\@tempdima}%
   \fi
   \ifvoid\L@tbox
     \sbox{\L@tbox}{\LettrineTextFont{\LettrineSecondString}}%
   \fi
   \addtolength{\L@height}{\ht\L@tbox}%
   \addtolength{\L@height}{\L@oversize\L@height}%
}
\newcommand*{\LettrineFontHook}{}
\newcommand*{\LettrineTestString}{ABCDEFGHIJKLMNOQPRSTUVWXYZ}
\newcommand*{\LettrineSecondString}{x}
\newcommand*{\LettrineFont}{%
   \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 the maximal capitals' height and compute a scaling factor (always ≥ 1). All capitals are expected to have the same height, in case this assumption would be wrong for some special font, `\LettrineTestString` can be customised to any non empty subset of capitals.

```
   \sbox{\@tempboxa}{\LettrineFontHook\fontsize{\L@height}{\L@height}%
                     \selectfont \LettrineTestString}%
```

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

```
   \@tempcntb=\ht\@tempboxa
   \@tempcnta=\L@height
   \multiply\@tempcnta by 100
   \divide\@tempcntb by 100
   \divide\@tempcnta by \@tempcntb
   \advance\@tempcnta by -9999
   \ifnum\@tempcnta>0
     \def\@tempa{1.\the\@tempcnta}%
   \else
     \def\@tempa{1}%
   \fi
   \LettrineFontHook
   \fontsize{\@tempa\L@height}{\@tempa\L@height}%
   \selectfont
}
```

`\LettrineFontEPS` 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.

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

Set up keyval initializations.

```
\define@key{L}{lines}{\setcounter{L@lines}{#1}}
\define@key{L}{depth}{\setcounter{L@depth}{#1}}
\define@key{L}{lhang}{\renewcommand*{\L@hang}{#1}}
\define@key{L}{loversize}{\renewcommand*{\L@oversize}{#1}}
\define@key{L}{lraise}{\renewcommand*{\L@raise}{#1}}
\define@key{L}{ante}{\renewcommand*{\L@ante}{#1}}
\define@key{L}{findent}{\setlength{\L@Findent}{#1}}
\define@key{L}{nindent}{\setlength{\L@Nindent}{#1}}
\define@key{L}{slope}{\setlength{\L@slope}{#1}}
\define@key{L}{image}[true]{\csname L@image#1\endcsname}
\define@key{L}{grid}[true]{\csname L@grid#1\endcsname}
\define@key{L}{realheight}[true]{\csname L@realh#1\endcsname}
\define@key{L}{novskip}{\setlength{\L@novskip}{#1}}
```

`\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’.

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

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

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

First reset the parameters to their default values :

```
  \setcounter{L@lines}{\value{DefaultLines}}%
  \setcounter{L@depth}{\value{DefaultDepth}}%
  \renewcommand*{\L@hang}{\DefaultLhang}%
  \renewcommand*{\L@oversize}{\DefaultLoversize}%
  \renewcommand*{\L@raise}{\DefaultLraise}%
  \renewcommand*{\L@ante}{}%
  \setlength{\L@Findent}{\DefaultFindent}%
  \setlength{\L@Nindent}{\DefaultNindent}%
  \setlength{\L@slope}{\DefaultSlope}%
```

```
\setlength{\L@novskip}{\DiscardVskip}%
\ifLettrineImage\L@imagetrue\else\L@imagefalse\fi
\ifLettrineOnGrid\L@gridtrue\else\L@gridfalse\fi
\ifLettrineRealHeight\L@realhtrue\else\L@realhfalse\fi
```

`\LettrineFont` and `\LettrineFontEPS` both call `\Lettrine@height` to set `\L@height` which depends on `\L@tbox`. The content of `\L@tbox` depends on option `realheight`, so we have to read the lettrine's options and initialise the `\L@tbox` content now[2].

```
\setkeys{L}{#1}%
\sbox{\L@tbox}{\LettrineTextFont{\LettrineSecondString}}%
\ifL@realh
  \def\@tempa{#3}
  \ifx\@tempa\@empty
    \PackageWarning{lettrine.sty}%
      {Empty second argument,\MessageBreak
       ignoring option `realheight';}%
  \else
    \sbox{\L@tbox}{\LettrineTextFont{#3}}%
  \fi
\fi
```

Then try to read an optional file (its name is given by `\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`)...

```
\if\DefaultOptionsFile\relax
\else
  \begingroup
  \InputIfFileExists{\DefaultOptionsFile}%
    {}%
    {\PackageWarning{lettrine.sty}%
       {File \DefaultOptionsFile\space not found}%
    }%
```

Gobble the colour commands, just keep the letter argument.

```
  \def\color##1##{\l@color{##1}}%
  \let\l@color\@gobbletwo
  \def\textcolor##1##{\l@textcolor{##1}}%
  \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).

```
  \expandafter\ifx\csname l@#2-keys\endcsname\relax
                \gdef\l@LOKeys{}%
              \else
                \xdef\l@LOKeys{\csname l@#2-keys\endcsname}%
              \fi
  \endgroup
```

[2] Now means before eventually reading the config file.

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

```
  \def\KV@prefix{KV@L@}%
  \let\@tempc\relax
  \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.

```
  \sbox{\L@lbox}{\LettrineFont #2}%
  \setlength{\LettrineWidth}{\wd\L@lbox}%
  \def\KV@prefix{KV@L@}%
  \let\@tempc\relax
  \expandafter\KV@do\l@LOKeys,\relax,
```

As local options prevail on those held in the config file, we have to read local options again :

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

Options and optional config file have be taken into account, we can now save the first mandatory argument of `\lettrine` properly scaled into `\L@lbox`. Depending on the boolean `image`, `\LettrineFont` or `\LettrineFontEPS` is used.

```
\ifL@image
   \sbox{\L@lbox}{\LettrineFontEPS{#2}}%
\else
   \sbox{\L@lbox}{\LettrineFont #2}%
\fi
```

Height calculations done, let's reset `\L@tbox`'s content (mandatory in case `realheight=false`) :

```
\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`.

```
\@tempdima=\baselineskip
\ifnum\value{L@lines}=1
  \setlength{\L@first}{\ht\L@lbox}%
  \addtolength{\L@first}{-\ht\L@tbox}%
  \setlength{\L@lraise}{\z@}%
\else
  \setlength{\L@first}{-\value{L@lines}\@tempdima}%
  \addtolength{\L@first}{\@tempdima}%
  \sbox{\@tempboxa}{\LettrineTextFont x}%
  \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`).

```
    \setlength{\L@lraise}{-\L@raise\L@first}%
    \addtolength{\L@first}{\L@lraise}%
    \addtolength{\L@first}{\ht\L@lbox}%
    \addtolength{\L@lraise}{-\value{L@lines}\@tempdima}%
    \addtolength{\L@lraise}{\@tempdima}%
  \fi
  \par
```

`\L@first` now holds the height of the needed `\vskip`; if too small it will be discarded.

```
  \ifdim\L@first>\L@novskip
```

When the 'grid' option is true, let's round up `\L@first` to the next integer number of `\baselineskip`.

```
    \ifL@grid
      \@tempdima=\baselineskip
      \@tempdimb=\@tempdima
      \advance\@tempdimb by \L@novskip
      \@tempcnta=1
      \loop\ifdim\L@first>\@tempdimb
         \advance\@tempcnta by 1
         \advance\L@first by -\@tempdima
      \repeat
      \vskip\@tempcnta\baselineskip
    \else
      \vskip\L@first
    \fi
  \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.

```
  \setlength{\L@Pindent}{\wd\L@lbox}%
  \addtolength{\L@Pindent}{-\L@hang\wd\L@lbox}%
  \settowidth{\L@first}{\L@ante}%
  \addtolength{\L@Pindent}{\L@first}%
  \addtolength{\L@Pindent}{\L@Findent}%
  \setlength{\L@first}{\linewidth}%
  \addtolength{\L@first}{-\L@Pindent}%
```

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

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

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

Store the lettrine's final dimensions :

```
  \setlength{\LettrineWidth}{\wd\L@lbox}%
  \setlength{\LettrineHeight}{\ht\L@lbox}%
  \setlength{\LettrineDepth}{\dp\L@lbox}%
```

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

```
  \addtocounter{L@lines}{1}%
  \addtocounter{L@lines}{\value{L@depth}}%
  \def\L@parshape{\c@L@lines \the\L@Pindent \the\L@first}%
  \@tempcnta=\tw@
  \@whilenum \@tempcnta<\c@L@lines\do{%
     \edef\L@parshape{\L@parshape \the\L@Nindent \the\L@next}%
     \addtolength{\L@Nindent}{\L@slope}%
     \addtolength{\L@next}{-\L@slope}%
     \advance\@tempcnta\@ne}%
  \edef\L@parshape{\L@parshape \rightmargin \the\linewidth}%
  \noindent
  \parshape=\L@parshape\relax
```

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

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

This ends the definition of `\lettrine`.

Load a local config file if present in LaTeX's search path.

```
\InputIfFileExists{lettrine.cfg}
   {\typeout{Loading lettrine.cfg}}
   {\typeout{lettrine.cfg not found, using default values}}
```

3 Configuration file

```
%% lettrine.cfg: configuration file for lettrine.sty
%%
%% If you want to customize lettrine, please *do not* hack into the
%% code, copy this file to the directory where lettrine.sty lies
%% and customize it as you like.
%%
%% Uncomment these lines and change the parameters' values to fit
%% your needs (see lettrine.dtx).
%%
%%\setcounter{DefaultLines}{2}
%%\setcounter{DefaultDepth}{0}
%%
%% These are *decimal* numbers:
%%\renewcommand*{\DefaultLoversize}{0}
%%\renewcommand*{\DefaultLraise}{0}
%%\renewcommand*{\DefaultLhang}{0}
%%
%% These are *lengths* (don't forget the unit):
```

```
%%\setlength{\DefaultFindent}{0pt}
%%\setlength{\DefaultNindent}{0.5em}
%%\setlength{\DefaultSlope}{0mm}
%%\setlength{\DiscardVskip}{0.2pt}
%%
%% Theses are *flags* (value=true/false):
%%\LettrineImagefalse
%%\LettrineOnGridfalse
%%\LettrineRealHeightfalse
%%
%% Theses are *commands* (value=string, only its height matters):
%%\renewcommand*{\LettrineTestString}{ABCDEFGHIJKLMNOQPRSTUVWXYZ}
%%\renewcommand*{\LettrineSecondString}{x}
%%
%% In case you want to set parameters for some letters
%% in file ‘optfile.cfl’
%%\renewcommand{\DefaultOptionsFile}{optfile.cfl}
```

Historique

lettrine-0.81

- `\lettrine` : `\DefaultLoversize` added. 7

lettrine-0.9

- `\LettrineFont` :
  - `\Lettrine@height` added. 6
  - `\LettrineFontHook` added. 6
  - Size of the dropped capital changed when ‘lines’ value is 1 (was `\Huge`). 6
- `\lettrine` : Calculations of length `\L@first` changed. Do not ‘vskip’ small lengths (<0.2pt), they are just rounding errors. . 9
- Général : `\LettrineFontEPS` added. 6

lettrine-1.1

- `\lettrine` : Add `\rightmargin` to `\L@Pindent` for `\Lettrine` to work properly in quote, quotation, abstract environments… but do not change `\linewidth` which is set by these environments. 10

lettrine-1.2

- `\LettrineFont` : `\baselineskip` may be a rubber length, we convert it to a dimen. 6
- `\lettrine` : `\baselineskip` may be a rubber length, we convert it to a dimen. 9
- Général : `\newlength` changed to `\newdimen`, to correct a bug with seminar.cls (pointed out by Peter Münster). 4

lettrine-1.3

- Général : Correct the documentation to mention the cm-super fonts and the type1ec package by Vladimir Volovich. . 3

lettrine-1.4

- `\lettrine` : `\lettrine` still didn’t work properly in quote, quotation, abstract environments, pointed out by Matthias C. Schmidt. `\rightmargin` was added too early to `\L@Nindent`, thus making `\Ł@next` too short by `\rightmargin`. 10

lettrine-1.5

- `\lettrine` : Added reading of an optional config file `\DefaultOptionsFile`. 8
- Général : `\LettrineOptionsFor` and `\LettrineWidth` added. . . 7

lettrine-1.6

- `\LettrineFontEPS` : Added `\LettrineFontHook` to `\LettrineFontEPS`. This is needed for color options. 7
- `\lettrine` : Add braces around #3 to allow commands taking an

argument (such as \MakeLowercase) in \LettrineTextFont. Suggested by Philipp Lehman. 9

Général : Add a flag to switch to images in eps or pdf format. Suggested by Bill Jetzer. 2

Added newif \ifL@grid. 5

Added newif \ifL@image. 5

Added newif \ifLettrineImage. 4

lettrine-1.63

\LettrineFont : Added command \LettrineTestString which defaults to 'ABCDEFGHIJKLMNOQPRSTUVWXYZ'. In previous versions height computations were based on letter 'X' which might not exist in some (rare) fonts. Pointed out by Raphaël Pinson. 6

lettrine-1.64

\lettrine : Remove $ around \smash and add \relax. Bug pointed out by David Monniaux. Correction by Enrico Gregorio. 11

lettrine-1.65

\lettrine : Measure and store the lettrine's final dimensions. . . . 10

lettrine-1.7

Général : New counter to add lines for dropped capitals with positive depth, like Q. 1

lettrine-1.8

\lettrine : The 0.2pt limit for discarded vskips is now customisable through \DiscardVskip and option 'novskip'. 9

Général : Added newif \ifLettrineOnGrid and new dimen \DiscardVskip, default (0.2pt) set for compatibility with previous releases. 4

Added two keyval options : 'grid' (true/false) and 'novskip' to override \DiscardVskip. . . . 2

lettrine-1.9

\LettrineFont : \theL@lines changed to \value{L@lines}. Needed for babel-hebrew which redefines \@arabic. 6

\lettrine : \theDefaultLines changed to \value{DefaultLines}, same with \theDefaultDepth. Needed for babel-hebrew which redefines \@arabic. Thanks to Ulrike Fischer for providing the fix. 7

\theL@depth changed to \value{L@depth}. 11

Use the second mandatory argument of \lettrine or \LettrineSecondString (which defaults to 'x') to compute \L@height. This is controlled by the 'realheight' flag. 8

Général : New customisable string \LettrineSecondString to tune the lettrine's height. 3

New keyval option : 'realheight' (true/false) and new global flag \ifLettrineRealHeight. 2