

Typesetting dropped capitals in L^AT_EX 2_ε documents

Daniel Flipo
daniel.flipo@free.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 Perrousseau, 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>` ($0 \leq \text{lhang} \leq 1$) 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);

¹The file described in this section has version number v2.22 and was last revised on 2019-04-06.

- `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`, `A.jpg`, `A.png` or `A.pdf` instead of letter A. This was suggested by Bill Jetzer.
N.B.: Redefining `\LettrineFont` as `\LettrineFontEPS` is no longer supported and `\LettrineFontEPS` has been renamed as `\LettrineImage`.
- `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 initial 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 ‘H’ 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 `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).

- `refstring`² (new to version 2.1) is meant for fancy initials with irregular heights (i.e. taken in fonts like Yinit (OpenType), `cfr-initials`,...). `refstring` forces the `\fontsize` computations to be run on the initial given as `\lettrine`’s first mandatory argument instead of the reference string

²Unlike `grid` or `realheight`, `refstring` is not a flag (*do not* add `=true!`); it is possible to set `refstring=<string>` to override `\LettrineTestString` locally.

`\LettrineTestString`. In most cases, this option should *not* be used: think of accentuated initials or capitals with optical correction.

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

Coloured initials are possible in conjunction 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 initials globally is described below, see `\LettrineFontHook`.

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

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`,
- `\LettrineSelfReffalse`,
- `\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 Z, 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 initials in a (part of) document: `\renewcommand{\LettrineFontHook}{\color[gray]{0.5}}` will colour the initials 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}`³ 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 issues:

- 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]`.

³This package, available on CTAN, was first released on 2002/07/30.

- `\lettrine` does not work within lists.
- If a *list* has to be included in a paragraph starting with `\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 L^AT_EX.
- If you are facing some slight height inaccuracy for a dropped capital, you can try option `refstring`; this option is meant for fancy (unaccented) initials. Informations about targeted and effective initial’s height are available in the `.log` file. Using LuaTeX or XeTeX engines with OpenType fonts may be an option (some TFM files for Type1 fonts are slightly inaccurate).
- `\LettrineTestString`’s value has changed over the time; these changes may result in slight size differences for the initial. Starting with version 2.2, the `lettrine` package takes advantage of the rollback facilities recently introduced by the LaTeX Team⁴. Three rollback versions are provided in order to produce exactly the same output as with former versions: you can request `\usepackage{lettrine}[=v1.6]`⁵ for 1999-2012 documents or `\usepackage{lettrine}[=v1.9]` (documents from 2012 to July 2018) or `\usepackage{lettrine}[=v2.0]` (August 2018). Using any date in ISO format works too: `\usepackage{lettrine}[=2014-03-15]` will load v1.9.

2 T_EXnical details

The `lettrine` package now uses the rollback mechanism recently introduced by the LaTeX Team to provide easier backward compatibility. This requires a recent LaTeX kernel (at least 2018-04-01), rollback is ignored with older kernels.

```

1 \ifdefined\DeclareRelease
2   \DeclareRelease{v1.6}{1999-03-03}{lettrine-2006-03-17.sty}
3   \DeclareRelease{v1.9}{2012-07-20}{lettrine-2015-08-31.sty}
4   \DeclareRelease{v2.0}{2018-07-21}{lettrine-2018-08-18.sty}
5   \DeclareCurrentRelease{}{2019-03-30}
6 \else
7   \PackageWarning{lettrine}{Your LaTeX kernel is too old to provide
8     access\MessageBreak to former versions of the lettrine package.%
9     \MessageBreak If you need rollback, please upgrade%
10    \MessageBreak your LaTeX kernel (2018-04-01 or newer),%
11    \MessageBreak otherwise you are fine; reported}
12 \fi

```

This package only runs with L^AT_EX_{2 ϵ} and requires files `keyval.sty` and `minifp.sty`.

```
13 \RequirePackage{keyval,minifp}
```

⁴A LaTeX kernel dated 2018-04-01 or newer is required.

⁵Don’t forget the = sign!

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`.

```

14 \newcounter{DefaultLines}
15 \setcounter{DefaultLines}{2}
16 \newcounter{DefaultDepth}
17 \newcommand*{\DefaultOptionsFile}{\relax}
18 \newcommand*{\DefaultLoversize}{0}
19 \newcommand*{\DefaultLraise}{0}
20 \newcommand*{\DefaultLhang}{0}
21 \newdimen\DefaultFindent
22 \setlength{\DefaultFindent}{\z@}
23 \newdimen\DefaultNindent
24 \setlength{\DefaultNindent}{0.5em}
25 \newdimen\DefaultSlope
26 \setlength{\DefaultSlope}{\z@}
27 \newdimen\DiscardVskip
28 \setlength{\DiscardVskip}{0.2\p@}
29 \newif\ifLettrineImage
30 \newif\ifLettrineOnGrid
31 \newif\ifLettrineRealHeight
32 \newif\ifLettrineSelfRef

```

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

```

33 \newsavebox{\L@lbox}
34 \newsavebox{\L@tbox}
35 \newcounter{L@lines}
36 \newcounter{L@depth}
37 \newdimen\L@Pindent
38 \newdimen\L@Findent
39 \newdimen\L@Nindent
40 \newdimen\L@lraise
41 \newdimen\L@first
42 \newdimen\L@next
43 \newdimen\L@slope
44 \newdimen\L@height
45 \newdimen\L@novskip
46 \newdimen\L@target@ht
47 \newdimen\L@target@dp
48 \newdimen\L@target@tht
49 \newdimen\LettrineWidth
50 \newdimen\LettrineHeight
51 \newdimen\LettrineDepth
52 \newcommand*{\L@file}{}
53 \newcommand*{\L@hang}{}
54 \newcommand*{\L@oversize}{}
55 \newcommand*{\L@raise}{}
56 \newcommand*{\L@ante}{}
57 \newif\ifL@image
58 \newif\ifL@grid
59 \newif\ifL@realh
60 \newif\ifL@selfref

```

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

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

```
61 \newcommand*{\LettrineTextFont}{\scshape}
62 \newcommand*{\LettrineSecondString}{x}
```

`\LettrineFontHook` `\LettrineFontHook` enables to select another font for the dropped capital. Its default definition is empty (the current text font is used).

```
63 \newcommand*{\LettrineFontHook}{}
```

`\computeL@height` 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 globally by redefining `\Defaultloversize` or locally using optional argument `loversize=`. If `lines=1`, the default size for the dropped capital is computed as if `lines=2`.

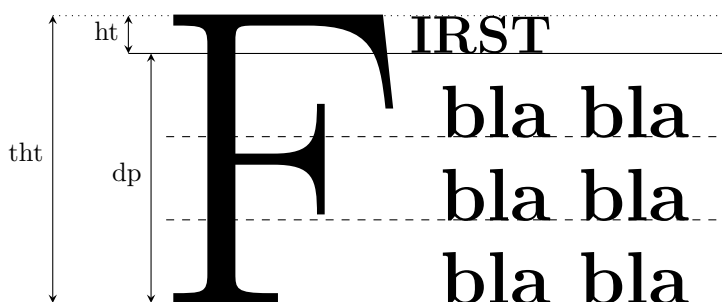


Figure 1: Initial's targeted dimensions (`\lines = 4`)

`\computeL@height` first computes the targeted height for the dropped capital and stores it into `\L@target@tth`. This height only depends on `L@lines` and on the height of `\L@tbox` (see fig. 1). So options *must* be read and `\L@tbox` must be properly initialised *before* executing `\computeL@height` (see below in `\@lettrine` code).

`\L@height` is set to `\L@target@tth` raised by the `\L@oversize` factor.

```
64 \def\computeL@height{%
65   \setlength{\L@target@ht}{\ht\L@tbox}%
```

As `\baselineskip` might be a rubber length, let's convert it into a 'dimen' using `\@tempdima`.

```
66   \@tempdima=\baselineskip
67   \setlength{\L@target@dp}{\value{L@lines}\@tempdima}%
68   \ifnum\value{L@lines}>1
69     \addtolength{\L@target@dp}{-\@tempdima}%
70   \else
71     \addtolength{\L@target@ht}{\L@target@dp}%
72     \setlength{\L@target@dp}{0pt}%
73   \fi
```

```

74 \setlength{\L@target@tth}{\L@target@ht}%
75 \addtolength{\L@target@tth}{\L@target@dp}%
76 \setlength{\L@height}{\L@target@tth}%
77 \addtolength{\L@height}{\L@oversize\L@target@tth}%
78 }

```

`\LettrineTestString` After executing `\computeL@height`, `\L@height` holds the exact height required for the dropped capital, nothing more is needed if the initial is a picture, otherwise we need to compute the matching `\fontsize`'s value. This is done by measuring the height of a “reference” capital (i.e. either listed in `\LettrineTestString` or the initial itself). As some font designers apply optical correction to capitals C, G, O, or Q (they are slightly taller than ‘T’ or ‘Z’), they are better left out of `\LettrineTestString`. EFTZ should be a good default for most fonts.

```

79 \newcommand*{\LettrineTestString}{EFTZ}

```

`\compute@fontsize` This command compares the height of a “reference” capital scaled by `\fontsize` with argument `\L@height` to `\L@height` (the required height for the initial); both are converted into integers (in sp) to compute a ratio `\L@factor`. Up to v2.01, possible values for `\L@factor` were either 1 or any value ≥ 1.1 which was fine for almost every font but not all (i.e. `cfr-initials`). Starting with v2.1, `\L@factor` is computed more accurately by `\MFPdiv` from `minifp.sty`.

When optional argument `selfref` is `true` the initial itself is taken as reference to compute `\fontsize`, this can be handy when working with fancy fonts (i.e. `cfr-initials`, `Yinit`). In most cases, the default is a better choice.

```

80 \def\compute@fontsize{%
81   \ifL@selfref
82     \def\Lettrine@RefString{\l@refstring}%
83   \else
84     \def\Lettrine@RefString{\LettrineTestString}%
85   \fi
86   \sbox{\@tempboxa}{\LettrineFontHook
87     \fontsize{\L@height}{\L@height}\selectfont
88     \Lettrine@RefString}%
89   \@tempcnta=\ht\@tempboxa
90   \ifnum\@tempcnta=0
91     \PackageWarning{lettrine}{Unable to compute \protect\fontsize!%
92     \MessageBreak \protect\LettrineTestString\space empty? reported}
93     \def\L@factor{1}%
94   \else
95     \@tempcntb=\L@height
96     \MFPdiv{\the\@tempcntb}{\the\@tempcnta}\L@factor
97   \fi
98 }

```

`\LettrineFont` `\fontsize`'s argument providing the requested `\L@height` is `\L@factor\L@height`.

```

99 \newcommand*{\LettrineFont}{%
100   \LettrineFontHook
101   \fontsize{\L@factor\L@height}{\L@factor\L@height}%
102   \selectfont
103 }

```


`\LettrineImage` 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 image at this size.

```
104 \newcommand*\LettrineImage{%
105   \LettrineFontHook\includegraphics[height=\L@height]%
106 }
```

`\setupL@lbox` The next (internal) command computes the requested size for the initial (letter or image) and prepares a box `\L@lbox` holding it.

```
107 \def\setupL@lbox{%
108   \computeL@height
109   \ifL@image
110     \sbox{\L@lbox}{\LettrineImage{\l@initial}}%
111   \else
112     \compute@fontsize
113     \sbox{\L@lbox}{\LettrineFont \l@initial}%
114   \fi
115 }
```

Set up keyval initializations.

```
116 \define@key{L}{lines}{\setcounter{L@lines}{#1}}
117 \define@key{L}{depth}{\setcounter{L@depth}{#1}}
118 \define@key{L}{lhang}{\renewcommand*\L@hang{#1}}
119 \define@key{L}{loversize}{\renewcommand*\L@oversize{#1}}
120 \define@key{L}{lraise}{\renewcommand*\L@raise{#1}}
121 \define@key{L}{ante}{\renewcommand*\L@ante{#1}}
122 \define@key{L}{findent}{\setlength{\L@Findent}{#1}}
123 \define@key{L}{nindent}{\setlength{\L@Nindent}{#1}}
124 \define@key{L}{slope}{\setlength{\L@slope}{#1}}
125 \define@key{L}{image}[true]{\csname L@image#1\endcsname}
126 \define@key{L}{grid}[true]{\csname L@grid#1\endcsname}
127 \define@key{L}{realheight}[true]{\csname L@realh#1\endcsname}
128 \define@key{L}{novskip}{\setlength{\L@novskip}{#1}}
129 \define@key{L}{refstring}[\l@initial]{\L@selfreftrue
130                                     \def\l@refstring{#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’.
```

```
131 \newcommand*\LettrineOptionsFor}[2]{\@namedef{l@#1-keys}{#2}}
```

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

```
132 \def\lettrine{\@ifnextchar[\@lettrine{\@lettrine[]}}
133 \def\@lettrine[#1]#2#3{%
134   \def\l@initial{#2}\let\l@refstring\l@initial
```

First reset the parameters to their default values:

```

135 \setcounter{L@lines}{\value{DefaultLines}}%
136 \setcounter{L@depth}{\value{DefaultDepth}}%
137 \renewcommand*{\L@hang}{\DefaultLhang}%
138 \renewcommand*{\L@oversize}{\DefaultLoversize}%
139 \renewcommand*{\L@raise}{\DefaultLraise}%
140 \renewcommand*{\L@ante}{}%
141 \setlength{\L@Findent}{\DefaultFindent}%
142 \setlength{\L@Nindent}{\DefaultNindent}%
143 \setlength{\L@slope}{\DefaultSlope}%
144 \setlength{\L@novskip}{\DiscardVskip}%
145 \ifLettrineImage\L@imagetrue\else\L@imagefalse\fi
146 \ifLettrineOnGrid\L@gridtrue\else\L@gridfalse\fi
147 \ifLettrineRealHeight\L@realhtrue\else\L@realhfalse\fi
148 \ifLettrineSelfRef\L@selfreftrue\else\L@selfreffalse\fi

```

The final initial size depends on the height of `\L@tbox`; the content of `\L@tbox` depends on option `realheight`, so we have to read `\lettrine`'s optional argument and initialise the `\L@tbox` content now⁶.

```

149 \setkeys{L}{#1}%
150 \sbox{\L@tbox}{\LettrineTextFont{\LettrineSecondString}}%
151 \ifL@realh
152   \def\@tempa{#3}%
153   \ifx\@tempa@empty
154     \PackageWarning{lettrine.sty}%
155       {Empty second argument,\MessageBreak
156         ignoring option `realheight';}%
157   \else
158     \sbox{\L@tbox}{\LettrineTextFont{#3}}%
159   \fi
160 \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 initial (defined by #2)...

```

161 \if\DefaultOptionsFile\relax
162 \else
163   \begingroup
164   \InputIfFileExists{\DefaultOptionsFile}%
165     {}%
166     {\PackageWarning{lettrine.sty}%
167       {File \DefaultOptionsFile\space not found}}%
168   }%

```

Gobble the colour commands, just keep the letter argument.

```

169 \def\color##1##{\l@color{##1}}%
170 \let\l@color@gobbletwo
171 \def\textcolor##1##{\l@textcolor{##1}}%
172 \def\l@textcolor##1##2##3{##3}%

```

⁶Now means before eventually reading the config file.

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

```

173   \expandafter\ifx\csname l@#2-keys\endcsname\relax
174       \gdef\l@LOKeys{}%
175   \else
176       \xdef\l@LOKeys{\csname l@#2-keys\endcsname}%
177   \fi
178   \endgroup

```

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

```

179   \def\KV@prefix{KV@L@}%
180   \let\@tempc\relax
181   \expandafter\KV@do\l@LOKeys,\relax,

```

As some parameters' values `findent`, `nindent` and `slope` — which do not influence the initial's size — may be given relative to `\LettrineWidth`, the `\L@lbox` has to be set up to evaluate `\LettrineWidth`.

```

182   \setupL@lbox
183   \setlength{\LettrineWidth}{\wd\L@lbox}%
184   \def\KV@prefix{KV@L@}%
185   \let\@tempc\relax
186   \expandafter\KV@do\l@LOKeys,\relax,

```

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

```

187   \setkeys{L}{#1}%
188   \fi

```

Options and optional config file have been taken into account, we can now finally save the first mandatory argument of `\lettrine` properly scaled into `\L@lbox`.

```

189   \setupL@lbox

```

Store the initial's final dimensions,

```

190   \setlength{\LettrineWidth}{\wd\L@lbox}%
191   \setlength{\LettrineHeight}{\ht\L@lbox}%
192   \setlength{\LettrineDepth}{\dp\L@lbox}%

```

print some informations about accuracy to the log file,

```

193   \begingroup
194   \def\IeC##1{##1}%
195   \@tempdima=\L@oversize pt\relax
196   \PackageInfo{lettrine.sty}%
197   {Targeted height = \the\L@target@tht\MessageBreak
198     (no accent, loversize=0),\MessageBreak
199     Lettrine height = \the\LettrineHeight\space (#2)%
200     \ifdim\@tempdima=z@\else\space loversize=\L@oversize\fi;%
201     \MessageBreak reported}%
202   \endgroup

```

and reset `\L@tbox`'s content (mandatory in case `realheight=false`):

```

203   \sbox{\L@tbox}{\LettrineTextFont{#3}}%

```

Start a new paragraph, skipping the necessary amount of space if the dropped capital sticks out of the top of paragraph. We use `\L@first` to compute the amount of space to be skipped.

The basis for `\L@raise` (and `\L@oversize`, see `\LettrineFont`) is `\L@target@tth`.

```

204 \setlength{\L@first}{\LettrineHeight}%
205 \setlength{\L@lraise}{\L@raise\L@target@tth}%
206 \addtolength{\L@first}{\L@lraise}%
207 \ifnum\value{L@lines}=1
208   \addtolength{\L@first}{-\ht\L@tbox}%
209 \else
210   \addtolength{\L@first}{-\L@target@tth}%
211   \addtolength{\L@lraise}{-\L@target@dp}%
212 \fi
213 \par

```

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

```

214 \ifdim\L@first>\L@novskip

```

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

```

215   \ifL@grid
216     \@tempdima=\baselineskip
217     \@tempdimb=\@tempdima
218     \advance\@tempdimb by \L@novskip
219     \@tempcnta=1
220     \loop\ifdim\L@first>\@tempdimb
221       \advance\@tempcnta by 1
222       \advance\L@first by -\@tempdima
223     \repeat
224     \L@first=\@tempcnta\baselineskip
225   \fi
226   \vskip\L@first
227 \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.

```

228 \setlength{\L@Pindent}{\wd\L@lbox}%
229 \addtolength{\L@Pindent}{-\L@hang\wd\L@lbox}%
230 \settowidth{\L@first}{\L@ante}%
231 \addtolength{\L@Pindent}{\L@first}%
232 \addtolength{\L@Pindent}{\L@Findent}%
233 \setlength{\L@first}{\linewidth}%
234 \addtolength{\L@first}{-\L@Pindent}%

```

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

```

235 \addtolength{\L@Nindent}{\L@Pindent}%
236 \setlength{\L@next}{\linewidth}%
237 \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 `\@totalleftmargin`.

```

238 \addtolength{\L@Pindent}{\@totalleftmargin}%
239 \addtolength{\L@Nindent}{\@totalleftmargin}%

```

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

```

240 \addtocounter{L@lines}{1}%
241 \addtocounter{L@lines}{\value{L@depth}}%
242 \def\L@parshape{\c@L@lines \the\L@Pindent \the\L@first}%
243 \@tempcnta=\tw@
244 \@whilenum \@tempcnta<\c@L@lines\do{%
245   \edef\L@parshape{\L@parshape \the\L@Nindent \the\L@next}%
246   \addtolength{\L@Nindent}{\L@slope}%
247   \addtolength{\L@next}{-\L@slope}%
248   \advance\@tempcnta\@ne}%
249 \edef\L@parshape{\L@parshape \@totalleftmargin \the\linewidth}%
250 \noindent
251 \parshape=\L@parshape\relax

```

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

```

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

```

A `\parshape` reset is required in abstract, quote and quotation environments beginning with `\lettrine` and spreading over several paragraphs. When the list ends, `\parshape` returns to 0.

```

255 \ifnum\@listdepth>0 \Lreset@listparshape \fi
256 }

```

This ends the definition of `\lettrine`; `\Lreset@listparshape` adds the `parshape` reset to the first occurrence of `\everypar` following the `\lettrine` command.

```

257 \newtoks\Llist@everypar
258 \def\Lreset@listparshape{%
259   \let\Lnew@everypar\everypar
260   \Llist@everypar=\expandafter{\the\everypar}%
261   \Lnew@everypar={\the\Llist@everypar
262     \parshape=\@ne \@totalleftmargin \linewidth \relax
263     \let\everypar\Llist@everypar
264     }%
265 }

```

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

```

266 \InputIfFileExists{lettrine.cfg}
267   {\typeout{Loading lettrine.cfg}}
268   {\typeout{lettrine.cfg not found, using default values}}

```

3 Configuration file

```
269 %% lettrine.cfg: configuration file for lettrine.sty
270 %%
271 %% If you want to customize lettrine, please *do not* hack into the
272 %% code, copy this file to your working directory and customize the
273 %% copy as you like.
274 %%
275 %% Uncomment any of these lines and change the parameters' values
276 %% to fit your needs (see lettrine.dtx).
277 %%
278 %%\setcounter{DefaultLines}{2}
279 %%\setcounter{DefaultDepth}{0}
280 %%
281 %% These are *decimal* numbers:
282 %%\renewcommand*\DefaultLoversize{0}
283 %%\renewcommand*\DefaultLraise{0}
284 %%\renewcommand*\DefaultLhang{0}
285 %%
286 %% These are *lengths* (don't forget the unit):
287 %%\setlength{\DefaultFindent}{0pt}
288 %%\setlength{\DefaultNindent}{0.5em}
289 %%\setlength{\DefaultSlope}{0mm}
290 %%\setlength{\DiscardVskip}{0.2pt}
291 %%
292 %% Theses are *flags* (value=true/false):
293 %%\LettrineImagefalse
294 %%\LettrineOnGridfalse
295 %%\LettrineRealHeightfalse
296 %%\LettrineSelfReffalse
297 %%
298 %% Theses are *commands* (value=string, only height matters):
299 %%\renewcommand*\LettrineTestString{EFTZ}
300 %%\renewcommand*\LettrineSecondString{x}
301 %%
302 %% In case you want to set parameters for some letters
303 %% in file `optfile.cfl'
304 %% \renewcommand{\DefaultOptionsFile}{optfile.cfl}
```

4 Change History

Changes are listed in reverse order (latest first) from version 1.0

v2.22	<code>\compute@fontsize</code> : Warn if <code>\fontsize</code> computation fails due to division by 0. 8		or X might be slightly taller (possible optical correction). . . . 8
	<code>\lettrine</code> : <code>\totalleftmargin</code> is the correct indentation for quote, quotation and abstract environments. 12		
	<code>\parshape</code> reset added in lists. 13		
v2.21	General: <code>\LettrineFontEPS</code> renamed as <code>\LettrineImage</code> . . . 9		
	Code clean up, new commands <code>\computeL@height</code> , <code>\compute@fontsize</code> , <code>\setupL@lbox</code> 7		
	<code>\lettrine</code> : <code>findent</code> computation relative to <code>\LettrineWidth</code> in <code>.cfl</code> files fixed; thanks to Frank Mittelbach for raising the issue. 11		
v2.2	General: Rollback mechanism used for recovering older versions. . . 5		
v2.1	General: New option ‘refstring’ and newif <code>\ifLettrineSelfRef</code> 6		
	Newif <code>\ifLettrineVone</code> and new option ‘Vone’ (removed in v2.2, rollback preferred). 6		
	<code>\compute@fontsize</code> : Computation of <code>\L@factor</code> for <code>\fontsize</code> done by the minifp package. . . . 8		
	<code>\computeL@height</code> : Height computations moved out of <code>\LettrineFont</code> : <code>\global</code> settings no longer required. . . . 7		
v2.0	<code>\computeL@height</code> : Store targeted dimensions of the dropped capital (ht, dp, tht) for further use. 7		
	<code>\lettrine</code> : Add informations about targeted and actual height of the initial to the <code>.log</code> file. 11		
	<code>\LettrineTestString</code> : changed from ‘ABCDEFGHIJKLMNO-QRSTUVWXYZ’ to ‘EFTZ’ as some capitals like C, G, O, Q		
		v1.9	General: New customisable string <code>\LettrineSecondString</code> to tune the initial’s height. 3
			New keyval option: ‘realheight’ (true/false) and new global flag <code>\ifLettrineRealHeight</code> 2
			<code>\computeL@height</code> : <code>\theL@lines</code> changed to <code>\value{L@lines}</code> . Needed for babel-hebrew which redefines <code>\@arabic</code> 7
			<code>\lettrine</code> : <code>\theDefaultLines</code> changed to <code>\value{DefaultLines}</code> , same with <code>\theDefaultDepth</code> . Needed for babel-hebrew which redefines <code>\@arabic</code> . Thanks to Ulrike Fischer for providing the fix. 10
			<code>\theL@depth</code> changed to <code>\value{L@depth}</code> 13
			Use the second mandatory argument of <code>\lettrine</code> or <code>\LettrineSecondString</code> (which defaults to ‘x’) to compute <code>\L@height</code> . This is controlled by the ‘realheight’ flag. 10
		v1.8	General: Added newif <code>\ifLettrineOnGrid</code> and new dimen <code>\DiscardVskip</code> , default (0.2pt) set for compatibility with previous releases. 6
			Added two keyval options: ‘grid’ (true/false) and ‘novskip’ to override <code>\DiscardVskip</code> 2
			<code>\lettrine</code> : The 0.2pt limit for discarded vskips is now customisable through <code>\DiscardVskip</code> and option ‘novskip’. 12
		v1.7	General: New counter to add lines for dropped capitals with positive depth, like Q. 1

v1.65	<code>\lettrine</code> : Measure and store the initial's final dimensions.	11	<code>\lettrine</code> : Added reading of an optional config file	
v1.64	<code>\lettrine</code> : Remove \$ around <code>\smash</code> and add <code>\relax</code> . Bug pointed out by David Monniaux. Correction by Enrico Gregorio.	13	<code>\DefaultOptionsFile</code>	10
v1.63	<code>\LettrineTestString</code> : (new) it defaults to 'ABCDEFGH-IJKLMNOPQRSTUVWXYZ'. In previous versions height computations were based on letter 'X' which might not exist in some (rare) fonts. Pointed out by Raphaël Pinson.	8	v1.4	<code>\lettrine</code> : <code>\lettrine</code> still didn't work properly in quote, quotation, abstract environments, pointed out by Matthias C. Schmidt.
v1.6	General: Add a flag to switch to images in eps or pdf format. Suggested by Bill Jetzer.	2	<code>\rightmargin</code> was added too early to <code>\L@Nindent</code> , thus making <code>\L@next</code> too short by <code>\rightmargin</code>	12
	Added newif <code>\ifL@grid</code>	6	v1.3	General: Correct the documentation to mention the cm-super fonts and the typelec package by Vladimir Volovich.
	Added newif <code>\ifL@image</code>	6	v1.2	General: <code>\newlength</code> changed to <code>\newdimen</code> , to correct a bug with seminar.cls (pointed out by Peter Münster).
	Added newif <code>\ifLettrineImage</code>	6	<code>\computeL@height</code> : <code>\baselineskip</code> may be a rubber length, we convert it to a dimen.	7
	<code>\lettrine</code> : Add braces around #3 to allow commands taking an argument (such as <code>\MakeLowercase</code>) in <code>\LettrineTextFont</code> . Suggested by Philipp Lehman.	11	<code>\lettrine</code> : <code>\baselineskip</code> may be a rubber length, we convert it to a dimen.	12
	<code>\LettrineImage</code> : Added <code>\LettrineFontHook</code> to <code>\LettrineFontEPS</code> . This is needed for color options.	9	v1.1	<code>\lettrine</code> : Add <code>\rightmargin</code> to <code>\L@Pindent</code> for <code>\Lettrine</code> to work properly in quote, quotation, abstract environments... but do not change <code>\linewidth</code> which is set by these environments.
v1.5	General: <code>\LettrineOptionsFor</code> and <code>\LettrineWidth</code> added.	9		