

# Kp-fonts: OpenType version

Daniel FLIPO

5th July 2022

This bundle provides OpenType versions of Type1 Kp-fonts designed by Christophe Caignaert. See `Kpfonts-Doc-English.pdf` for the full documentation of the original Type1 fonts.

It is usable only with LuaTeX or XeTeX engines; it consists of sixteen Text OpenType fonts, a Roman family **KpRoman** in eight shapes and weights—*Regular, Italic, Bold, BoldItalic, Light, LightItalic, Semibold, SemiboldItalic*—, a Sans-Serif family **KpSans** and a TypeWriter family **KpMono**, each of them in four shapes and weights—*Regular, Italic, Bold* and *BoldItalic*— and five Math OpenType fonts **KpMath-Regular**, **KpMath-Bold**, **KpMath-Light**, **KpMath-Semibold** and **KpMath-Sans**.

**KpRoman** and **KpSans** families have small caps available in two sizes (SmallCaps and PetiteCaps), upper and lowercase digits (0123456789), ancient ligatures  $\text{ct}$ ,  $\text{st}$  and  $\text{Q}$  a long-tailed capital Q. Superior and inferior digits and letters have been added to the OpenType **KpRoman** and **KpSans** fonts for footnotes' calls and abbreviations 1<sup>st</sup>, 2<sup>nd</sup>...

The support of text fonts covers Latin and Latin Extended A (U+0020 to U+017F) but neither IPA nor Greek nor Cyrillic alphabets.

For all Math fonts, Latin and Greek letters are available in Upright and Italic shapes, in Bold and Regular weights:  $\alpha \beta \text{C} \Delta$ ,  $\alpha \beta \text{C} \Delta$ ,  $\mathbf{\alpha} \mathbf{\beta} \mathbf{C} \mathbf{\Delta}$ ,  $\mathbf{\alpha} \mathbf{\beta} \mathbf{C} \mathbf{\Delta}$ .

Blackboard Bold capitals are available in two shapes, Serif and Sans: `\mathbb{ABC}` prints either  $\mathbb{A}\mathbb{B}\mathbb{C}$  (default) or  $\mathbb{A}\mathbb{B}\mathbb{C}$  (option `[Style=bbsans]`) Commands `\mathcal{ABC}` and `\mathscr{ABC}` print respectively  $\mathcal{ABC}$  and  $\mathscr{ABC}$  while `\mathfrak{ABCabc}` prints  $\mathfrak{A}\mathfrak{B}\mathfrak{Cabc}$ .

File `unimath-kpfonts.pdf` shows the full list of Unicode Math symbols provided by Kp-fonts, compared with other common Math fonts. More symbols, specific to Kp-fonts, are listed in section 3.2.

A style file `kpfont-otf.sty` is provided to load Kp-fonts easily. It is derived from `kpfont.sty` but options differ.

Please beware of the *experimental* status of the current version (0.43).

All fonts are covered by OFL licence, style file and documentation are under LPPL-1.3 licence.



**oldstylenums:** provides lowercase digits as a default.

To get uppercase digits locally: `{\addfontfeature{Numbers=Lining} 123}`.

Examples, upright, italic, bold and bolditalic:

- 0123456789!
- *0123456789!*
- **0123456789!**
- ***0123456789!***

**oldstyle:** provides lowercase digits as a default, long-tailed Q (Quebec) and (for Roman and Sans-Serif fonts only) old style ligatures « ct » et « st ».

Examples:

- *Quest* for an attractive font!
- *Quest* for an attractive font!
- **Quest** for an attractive font!
- **QUEST** FOR AN ATTRACTIVE FONT!
- *Quest* for an attractive font!
- **QUEST** FOR AN ATTRACTIVE FONT!

**veryoldstyle:** same as option `oldstyle` but the round ‘s’ is replaced by the long one ‘ſ’, unless it ends a word (then it remains ‘s’)<sup>2</sup> and ancient ligatures fi, fl, ft are activated. Coding `s=` prints a round ‘s’ anytime; in most cases this coding is not necessary with LuaTeX, f.i. `\textit{some of Highlands’ mysterious castles...}` will print *some of Highlands’ mysterious castles...* which is correct; with XeTeX an = sign must be added at the end of `mysterious=` to get a round ‘s’ there : final ‘s’ followed by a punctuation sign is never turned into ‘ſ’, when followed by a space it is a trickier, see option `longs` below.

**longs** inhibits the transformation of an ‘s’ followed by a space into ‘ſ’. This option, available only with LuaTeX, is silently switched on by `veryoldstyle`; it is used explicitly in the present document to deal with the examples of long ‘s’ shown in the preceding item.

**largesmallcaps:** prints larger SMALL CAPS than the default ones (PETITE CAPS).

**altfligs:** prints alternative shapes for ligatures fi, fl, ffi, ffl instead of fi, fl, ffi, ffl.

**germandbls:** `\SS` prints SS instead of ß (capital *Eszett*), ditto for small/petite caps.

**eurosym:** replaces the Euro character (€) by the official symbol (€) (available in regular, italic, bold and et bolditalic).

**harfbuzz:** switches `Renderer=Harfbuzz` for HBLuaTeX engine; up to version 0.34, this renderer was silently activated but seldom useful.

### 1.3 Options for Math fonts *only*

**lightmath:** uses *light* Math fonts.

---

2. The OpenType `calt` feature is used to catch ‘s’ ending a word.

**bbsans:** command `\mathbb` prints Sans-Serif Blackboard Bold capitals with Serif fonts too: compare  $\mathbb{C}, \mathbb{K}, \mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{Z}$ , with  $\mathbf{C}, \mathbf{K}, \mathbf{N}, \mathbf{Q}, \mathbf{R}, \mathbf{Z}$  (default).

**frenchstyle:** Latin uppercase letters and all Greek letters are printed upright, only lowercase latin letters are printed in italics; this follows the French typographic usage.

**oldstylenumsmath:** prints lowercase digits in Maths (default is uppercase).

**narrowiints:** prints condensed repeated integrals :

$\int\int\int$  et  $\iiint$  instead of  $\iiint$  et  $\iiint$  (default).

**partialup:** the `\partial` symbol is printed upright  $\partial$  instead of  $\partial$ .

**fancyReIm:** commands `\Re` et `\Im` print  $\Re$  and  $\Im$  respectively instead of  $\mathbb{R}$  et  $\mathbb{I}$ .

**tight:** horizontal spaces tightened in math mode (same settings as `fourier-otf`).

**noDcommand:** do not define `\D` to avoid incompatibilities with other packages.

Please note that the `mathcal` option has been deleted: commands `\mathcal{ABC}` and `\mathscr{ABC}` now print  $ABC$  and  $\mathcal{ABC}$  respectively when `kpfonts-otf.sty` is loaded.

## 2 Another way to load Kp-fonts

Loading Kp-fonts through `kpfonts-otf.sty` offers only a limited choice of options; the standard commands `\setmainfont`, `\setsansfont`, `\setmonofont`, `\setmathfont` offer much more flexibility.

On the other hand, `kpfonts-otf.sty` defines a lot of useful commands to access AMS and specific Kp-fonts symbols. Loading `kpfonts-otf` with the `symbols` option enables to get all these commands defined without loading any font:

```
\usepackage[symbols]{kpfonts-otf}
```

Please note that `unicode-math`<sup>3</sup> (and `fontspec`) *are loaded* by this procedure, no need to do it again, unless specific options are required, then `unicode-math` has to be loaded *before* `kpfonts-otf`, f.i.:

```
\usepackage[math-style=ISO,bold-style=upright]{unicode-math}
\usepackage[symbols]{kpfonts-otf}
```

Then, it is up to the user to load Kp-fonts with whatever option he/she likes using commands

```
\set...font{font}[options].
```

For documents requiring no Math fonts, loading `fontspec` and using the `\set...font` commands is enough, no need to load `kpfonts-otf` at all.

---

3. A carefull reading of both manuals `unicode-math.pdf` and `fontspec.pdf` (available in all TeX distributions) is required in order to take full advantage of these packages.

## 2.1 Options for Text fonts

Here are the options available for Text Kp-fonts:

**Numbers=Lowercase** to get lowercase digits 1,2,3 instead of 1,2,3; the default is **Numbers=Lining**.

**SmallCapsFeatures = {Letters=SmallCaps}** the `\textsc{}` command will print larger SMALL CAPS than the default PETITE CAPS.

The default setting<sup>4</sup>, is **SmallCapsFeatures = {Letters=PetiteCaps}**.

**Ligatures=TeX** (default) ' !` ?` -- --- print respectively ' i ÿ - —.

**Ligatures=Common** (default) automatic ligatures ff ffi ffl fi fl.

**StylisticSet=1** provides an alternative for glyphs ffi ffl fi fl (ff is unchanged).

**Ligatures=Required** : adds ft et tt ligatures.

**Ligatures=Rare** adds çt et st ligatures.

**Style=Swash** to get the long-tailed capital Q: Queen, also in small caps (both sizes): QUEEN and QUEEN.

**Style=Historic** replaces any instance of 's' by the long variant f. It is still possible to get a round 's' coding it as 's=' specially at end of words; see options `veryoldstyle` and `longs` p. 3 for more details.

**Ligatures=Historic** switches specific ligatures for the long f: fi, fl, ft.

**StylisticSet=2** : \SS prints SS instead of ß (capital *Eszett*), ditto for small/petite caps.

**StylisticSet=3** : replaces the Euro character (€) by the official symbol (€) (available in regular, italics, bold and et bolditalic).

Options may be are chosen for each font, say:

```
\setmonofont{KpMono}[Numbers=Lowercase,Style=Historic]
```

but can also be shared by different typefaces:

```
\defaultfontfeatures+[KpRoman,KpSans,KpMono]{Numbers=Lowercase}
\defaultfontfeatures+[KpRoman,KpSans]{%
  Ligatures = Rare,
  Style = Swash,
  SmallCapsFeatures = {Letters=PetiteCaps},
}
\setmainfont{KpRoman}
\setsansfont{KpSans}
\setmonofont{KpMono}
```

Notes : 1. `\setmonofont{KpMono}`, `\setsansfont{KpSans}`, `\setmainfont{KpRoman}` rely on files `KpMono.fontspec`, `KpSans.fontspec` and `KpRoman.fontspec` installed by Kpfonts.

---

4. Changed in v0.37 to match the original kpfonts package.



Both lines are mandatory: the first one loads **KpMath** as usual while the second one modifies `\mathcal{}` command's output.

The next four tables present the other Style Variants available:

Table 1 – Style=leqslant (+ss04)

| Command                  | Default       | Variant       |
|--------------------------|---------------|---------------|
| <code>\leq</code>        | $\leq$        | $\leqslant$   |
| <code>\geq</code>        | $\geq$        | $\geqslant$   |
| <code>\nleq</code>       | $\nleq$       | $\nleqslant$  |
| <code>\ngeq</code>       | $\ngeq$       | $\ngeqslant$  |
| <code>\leqq</code>       | $\leqq$       | $\leqslant$   |
| <code>\geqq</code>       | $\geqq$       | $\geqslant$   |
| <code>\leqless</code>    | $\leqless$    | $\leqless$    |
| <code>\eqgtr</code>      | $\eqgtr$      | $\eqgtr$      |
| <code>\lesseqgtr</code>  | $\lesseqgtr$  | $\lesseqgtr$  |
| <code>\gtreqless</code>  | $\gtreqless$  | $\gtreqless$  |
| <code>\lesseqqgtr</code> | $\lesseqqgtr$ | $\lesseqqgtr$ |
| <code>\gtreqqless</code> | $\gtreqqless$ | $\gtreqqless$ |

Table 2 – Style=smaller (+ss05)

| Command                      | Default      | Variant      |
|------------------------------|--------------|--------------|
| <code>\mid</code>            | $ $          | $\! $        |
| <code>\nmid</code>           | $\! $        | $\! $        |
| <code>\parallel</code>       | $\parallel$  | $\parallel$  |
| <code>\nparallel</code>      | $\nparallel$ | $\nparallel$ |
| <code>\parallelslant</code>  | $\parallel$  | $\parallel$  |
| <code>\nparallelslant</code> | $\nparallel$ | $\nparallel$ |

Table 3 – Style=subsetneq (+ss06)

| Command                  | Default       | Variant       |
|--------------------------|---------------|---------------|
| <code>\subsetneq</code>  | $\subsetneq$  | $\subsetneq$  |
| <code>\supsetneq</code>  | $\supsetneq$  | $\supsetneq$  |
| <code>\subsetneqq</code> | $\subsetneqq$ | $\subsetneqq$ |
| <code>\supsetneqq</code> | $\supsetneqq$ | $\supsetneqq$ |

Table 4 – Style=parallelslant (+ss07)

| Command                      | Default      | Variant      |
|------------------------------|--------------|--------------|
| <code>\parallel</code>       | $\parallel$  | $\parallel$  |
| <code>\nparallel</code>      | $\nparallel$ | $\nparallel$ |
| <code>\shortparallel</code>  | $\parallel$  | $\parallel$  |
| <code>\nshortparallel</code> | $\nparallel$ | $\nparallel$ |

Example: switching styles 4 (leqslant) and 6 (subsetneq) can be achieved coding either `\setmathfont{KpMath-Regular.otf}[StylisticSet={4,6}]` or `\setmathfont{KpMath-Regular.otf}[Style={leqslant,subsetneq}]` but this second syntax is available only if `kpfonts-otf.sty` has been loaded (eventually with the `symbols` option).

Table 5 on the following page shows the available ‘Glyphs Variants’:

Example: with `\setmathfont{KpMath-Regular.otf}[CharacterVariant={3,6}]` commands `\epsilon` and `\phi` print  $\epsilon$  and  $\phi$  instead of  $\epsilon$  et  $\phi$ . The same is true of course for all shapes and and weights (upright, bold, bolditalic, sans-serif, etc.): f.i. with `math-style=french`, `\epsilon` and `\phi` print  $\epsilon$  and  $\phi$  (upright).

Note about `\hbar`: `unicode-math` defines `\hbar` as `\hslash` (U+210F) while `amsmath` provides two different glyphs (italic h with horizontal or diagonal stroke). `kpfonts-otf` now follows `unicode-math`; the italic h with horizontal stroke can be printed using `\hslash` or `\hbar` together with character variant `cv01` or with `\mathbar` (replacement for AMS' command `\hbar`).

Table 5 – Glyphs Variants

|      | Default     | Variant     | Command                                    |
|------|-------------|-------------|--|
| cv00 | $\Re$ $\Im$ | $\Re$ $\Im$ | <code>\Re</code> <code>\Im</code>          |
| cv01 | $\hbar$     | $\hbar$     | <code>\hslash</code> or <code>\hbar</code> |
| cv02 | $\emptyset$ | $\emptyset$ | <code>\emptyset</code>                     |
| cv03 | $\epsilon$  | $\epsilon$  | <code>\epsilon</code>                      |
| cv04 | $\kappa$    | $\kappa$    | <code>\kappa</code>                        |
| cv05 | $\pi$       | $\pi$       | <code>\pi</code>                           |
| cv06 | $\phi$      | $\phi$      | <code>\phi</code>                          |
| cv07 | $\rho$      | $\rho$      | <code>\rho</code>                          |
| cv08 | $\sigma$    | $\sigma$    | <code>\sigma</code>                        |
| cv09 | $\theta$    | $\theta$    | <code>\theta</code>                        |
| cv10 | $\Theta$    | $\Theta$    | <code>\Theta</code>                        |

### 3 Specific commands defined in `kpfonts-otf.sty`

#### 3.1 Integrals

Kp-fonts offers variants for integral symbols suitable for indefinite integrals, they are coded as `\varint`, `\variint`, `\variiint`, `\variiiiint` and `\varidotsint`. Compare  $\int f(t) dt$  and  $\int f(t) dt$  and also

$$\int f(t) dt \quad \text{and} \quad \int f(t) dt$$

`\D{...}` prints an upright ‘d’ and improves kernings around the differential element: `\displaystyle\varint f(t)\D{t}` prints  $\int f(t) dt$ .

#### 3.2 Specific Math symbols

The next tables present symbols unavailable as Unicode characters, they are coded in Kp-fonts’ private zone.

|                                  |                      |                              |                     |
|----------------------------------|----------------------|------------------------------|---------------------|
| <code>\mmapsto</code>            | $\mapsto$            | <code>\longmmapsto</code>    | $\mapsto$           |
| <code>\mmapsfrom</code>          | $\mapsfrom$          | <code>\longmmapsfrom</code>  | $\mapsfrom$         |
| <code>\Mmapsto</code>            | $\Mmapsto$           | <code>\Longmmapsto</code>    | $\Mmapsto$          |
| <code>\Mmapsfrom</code>          | $\Mmapsfrom$         | <code>\Longmmapsfrom</code>  | $\Mmapsfrom$        |
| <code>\leftrightdasharrow</code> | $\leftrightarrow$    | <code>\leadsto</code>        | $\leadsto$          |
| <code>\boxright</code>           | $\boxrightarrow$     | <code>\boxleft</code>        | $\boxleftarrow$     |
| <code>\circleright</code>        | $\circrightarrow$    | <code>\circleleft</code>     | $\circleftarrow$    |
| <code>\Diamondright</code>       | $\diamondrightarrow$ | <code>\Diamondleft</code>    | $\diamondleftarrow$ |
| <code>\boxdotright</code>        | $\boxdotrightarrow$  | <code>\boxdotleft</code>     | $\boxdotleftarrow$  |
| <code>\circledotright</code>     | $\circledrightarrow$ | <code>\circledotleft</code>  | $\circledleftarrow$ |
| <code>\Diamonddotright</code>    | $\diamondrightarrow$ | <code>\Diamonddotleft</code> | $\diamondleftarrow$ |



|                                   |                   |                                   |                   |
|-----------------------------------|-------------------|-----------------------------------|-------------------|
| $\backslash\boxRight$             | $\Rightarrow$     | $\backslash\boxLeft$              | $\Leftarrow$      |
| $\backslash\boxdotRight$          | $\Rightarrow$     | $\backslash\boxdotLeft$           | $\Leftarrow$      |
| $\backslash\DiamondRight$         | $\Leftrightarrow$ | $\backslash\DiamondLeft$          | $\Leftrightarrow$ |
| $\backslash\DiamonddotRight$      | $\Leftrightarrow$ | $\backslash\DiamonddotLeft$       | $\Leftrightarrow$ |
|                                   |                   |                                   |                   |
| $\backslash\multimapdot$          | $\multimap$       | $\backslash\multimapdotinv$       | $\multimap$       |
| $\backslash\multimapdotboth$      | $\multimap$       | $\backslash\multimapdotbothvert$  | $\multimap$       |
| $\backslash\multimapbothvert$     | $\multimap$       | $\backslash\multimapdotbothBvert$ | $\multimap$       |
| $\backslash\multimapdotbothAvert$ | $\multimap$       | $\backslash\multimapdotbothBvert$ | $\multimap$       |
|                                   |                   |                                   |                   |
| $\backslash\capplus$              | $\cap$            | $\backslash\sqcupplus$            | $\sqcup$          |
| $\backslash\parallelslant$        | $\parallel$       | $\backslash\colonsim$             | $\sim$            |
| $\backslash\parallelbackslant$    | $\parallel$       | $\backslash\Colonsim$             | $\sim$            |
| $\backslash\eqqColon$             | $\equiv$          | $\backslash\Colondash$            | $\dashv$          |
| $\backslash\sqcapplus$            | $\sqcap$          | $\backslash\sqcupplus$            | $\sqcup$          |
| $\backslash\colonapprox$          | $\approx$         | $\backslash\Colonapprox$          | $\approx$         |
| $\backslash\dashColon$            | $\dashv$          | $\backslash\Colondash$            | $\dashv$          |
|                                   |                   |                                   |                   |
| $\backslash\strictif$             | $\rightarrow$     | $\backslash\strictfi$             | $\rightarrow$     |
| $\backslash\circledvee$           | $\circledvee$     | $\backslash\circledwedge$         | $\circledwedge$   |
| $\backslash\openJoin$             | $\times$          | $\backslash\opentimes$            | $\times$          |
| $\backslash\lambdaslash$          | $\lambda$         | $\backslash\lambdabar$            | $\lambda$         |
| $\backslash\strictiff$            | $\rightarrow$     | $\backslash\strictiff$            | $\rightarrow$     |
| $\backslash\circledbar$           | $\circledbar$     | $\backslash\circledbar$           | $\circledbar$     |
| $\backslash\VvDash$               | $\Vdash$          | $\backslash\Wr$                   | $\wr$             |
| $\backslash\Wr$                   | $\wr$             | $\backslash\Wr$                   | $\wr$             |
|                                   |                   |                                   |                   |
| $\backslash\idotsint$             | $\int \dots \int$ | $\backslash\idotsint$             | $\int \dots \int$ |
| $\backslash\ointclockwise$        | $\oint$           | $\backslash\varointctrlockwise$   | $\oint$           |
| $\backslash\oiintclockwise$       | $\oiint$          | $\backslash\oiintctrlockwise$     | $\oiint$          |
| $\backslash\varoiintclockwise$    | $\varoiint$       | $\backslash\varoiintctrlockwise$  | $\varoiint$       |
| $\backslash\oiiintclockwise$      | $\oiiint$         | $\backslash\oiiintctrlockwise$    | $\oiiint$         |
| $\backslash\varoiiintclockwise$   | $\varoiiint$      | $\backslash\varoiiintctrlockwise$ | $\varoiiint$      |
| $\backslash\sqiint$               | $\sqiint$         | $\backslash\sqiint$               | $\sqiint$         |
| $\backslash\sqiiiint$             | $\sqiiiint$       | $\backslash\sqiiiint$             | $\sqiiiint$       |

The full list of Unicode symbols available with Kp-fonts is shown in file `unimath-kpfonts.pdf`.

### 3.3 Wide accents

— `\widehat` and `\widetilde`

$\hat{x}$   $\widehat{xx}$   $\widehat{xxx}$   $\widehat{xxxx}$   $\widehat{xxxxx}$   $\widehat{xxxxxx}$   $\tilde{x}$   $\widetilde{xx}$   $\widetilde{xxx}$   $\widetilde{xxxx}$   $\widetilde{xxxxx}$   $\widetilde{xxxxxx}$

— `\overline` and `\underline`

$$\bar{x} \quad \overline{xy} \quad \overline{xyz} \quad \overline{A \cup B} \quad \overline{A \cup (B \cap C) \cup D} \quad \underline{m+n+p}$$

— `\wideoverbar`, `\widecheck` et `\widebreve`

$$\bar{x} \quad \overline{xy} \quad \overline{xyz} \quad \check{x} \quad \overline{\overline{xxxx}} \quad \overline{\overline{\overline{xxxxxx}}} \quad \check{x} \quad \overline{\overline{xxxx}} \quad \overline{\overline{\overline{xxxxxx}}}$$

— `\overparen` and `\underparen`

$$\widehat{x} \quad \widehat{xy} \quad \widehat{xyz} \quad \overset{\circ}{A \cup B} \quad \overset{\circ}{A \cup (B \cap C) \cup D} \quad \frac{2}{x+y} \quad \overset{26}{a+b+\dots+z}$$

$$\underline{x} \quad \underline{xz} \quad \underline{xyz} \quad \frac{x+z}{2} \quad \frac{a+b+\dots+z}{26}$$

— `\overbrace` and `\underbrace`

$$\overbrace{a} \quad \overbrace{ab} \quad \overbrace{abc} \quad \overbrace{abcd} \quad \overbrace{abcde} \quad \overbrace{a+b+c}^3 \quad \overbrace{a+b+\dots+z}^{26}$$

$$\underbrace{a} \quad \underbrace{ab} \quad \underbrace{abc} \quad \underbrace{abcd} \quad \underbrace{abcde} \quad \underbrace{a+b+c}_3 \quad \underbrace{a+b+\dots+z}_{26}$$

— `\overrightarrow` and `\overleftarrow`

$$\vec{v} \quad \vec{M} \quad \vec{vv} \quad \vec{AB} \quad \vec{ABC} \quad \vec{ABCD} \quad \vec{ABCDEFGH}$$

$$\overleftarrow{v} \quad \overleftarrow{M} \quad \overleftarrow{vv} \quad \overleftarrow{AB} \quad \overleftarrow{ABC} \quad \overleftarrow{ABCD} \quad \overleftarrow{ABCDEFGH}$$

— Enfin `\widearc` and `\widearccarrow` (ou `\overrightarrowarc`)

$$\widehat{AMB} \quad \widehat{AMB}$$

### 3.4 Math Versions

Different versions of the **KpMath** fonts may be defined in the document's preamble:

```
\setmathfont{KpMath-Regular.otf}[version=base, options ]
```

```
\setmathfont{KpMath-Bold.otf}[version=bold, options ]
```

```
\setmathfont{KpMath-Semibold.otf}[version=semibold, options ]
```

```
\setmathfont{KpMath-Sans.otf}[version=sans, options ]
```

```
\setmathfont{KpMath-Light.otf}[version=light, options ]
```

then, it is easy to switch from one version to another one with `\mathversion{name}`.

Example <sup>6</sup> :

```
\setmathfont{KpMath-Regular.otf}[Style=leqslant, CharacterVariant=3]  
\setmathfont{KpMath-Bold.otf}[version=bold,  
Style=leqslant, CharacterVariant=3]  
\setmathfont{KpMath-Sans.otf}[version=sans,  
Style=leqslant, CharacterVariant=3]
```

Here is the same equation in three versions, normal, bold and Sans-Serif:

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

`\mathversion{bold}`

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

`\mathversion{sans}`

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

---

6. Option `CharacterVariant=3` changes  $\epsilon$  into  $\varepsilon$ .