

# Kp-fonts: OpenType version

Daniel FLIPO

2nd May 2024

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 six maths OpenType fonts **KpMath-Regular**, **KpMath-Bold**, **KpMath-Light**, **KpMath-Semibold**, **KpMath-Sans** and **KpMath-SansBold**.

**KpRoman** and **KpSans** families have small caps available in two sizes (SmallCaps and PetiteCaps), upper and lowercase digits (0123456789), ancient ligatures *ct*, *st* and *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 maths fonts, Latin and Greek letters are available in Upright and Italic shapes, in Bold and Regular weights:  $\alpha \beta \gamma \delta$ ,  $\alpha \beta \gamma \delta$ ,  $\alpha \beta \gamma \delta$ ,  $\alpha \beta \gamma \delta$ .

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  $\mathcal{A}\mathcal{B}\mathcal{C}$  and  $\mathscr{A}\mathscr{B}\mathscr{C}$  while `\mathfrak{ABCabc}` prints  $\mathfrak{A}\mathfrak{B}\mathfrak{C}\mathfrak{a}\mathfrak{b}\mathfrak{c}$ .

File `unimath-kpfonts.pdf` shows the full list of Unicode maths symbols provided by Kp-fonts, compared with other common maths 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 (0.63).

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

# 1 Loading `kpfonts-otf.sty`

For users of the original `kpfonts.sty` package, the easiest way to try the OpenType version is to load `kpfonts-otf.sty`:

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

this loads `unicode-math`, `fontspec` and usually `realscripts`<sup>1</sup>, and defines **KpRoman** (Regular or Light depending on options), **KpSans** and **KpMono** as Text fonts, **KpMath** (Regular or Light depending on options) as maths fonts.

`kpfonts-otf.sty` also defines all symbols available in `lathexsym` and `amssymb` under the same names<sup>2</sup> and a bunch of Kp-fonts specific symbols.

## 1.1 Global options for both text and maths

**light:** switches to *light* fonts, metrics are unchanged; *light* fonts might not look perfect on screen but they print fine.

Please compare *regular* (left) and *light* fonts (right):

Normal or light? Just a matter of taste. Normal or light? Just a matter of taste.

$E = mc^2$

$E = mc^2$

*Normal or light? Just a matter of taste*    *Normal or light? Just a matter of taste*

**Normal or light? Just a matter of taste**    **Normal or light? Just a matter of taste**

*Normal or light? Just a matter of taste*    *Normal or light? Just a matter of taste*

**sans:** switches to *sans-serif* fonts for text and maths.

**nomath:** load neither `unicode-math` nor **KpMath** fonts; useful for documents without maths, or to choose other maths fonts.

**notext:** do not change any Text font, use the defaults.

**nosf:** do not change Sans-Serif Text fonts, use the defaults.

**nott:** do not change Typewriter Text fonts, use the defaults.

**onlyrm:** equivalent to the last two combined.

**fulloldstyle:** equivalent to options `oldstyle` and `oldstylemath`.

**fulloldstylenums:** equivalent to options `oldstylenums` and `oldstylenumsmath`.

## 1.2 Options for text fonts *only*

**lighttext:** switches to *light* Text fonts.

Two more weights are provided by `kpfonts-otf.sty`: with the *lighttext* (or *light*) option, *Semibold* and *Extrabold* vs *Light* and *Semibold* without it. These weights are

---

1. See option `fakedscripts` below.

2. Unicode names often differ from AMS names.

available through `\ltseries`, `\sbseries` and `\ebseries` commands to be used in a group or alternatively through one argument commands `\textlt{}`, `\textsb{}` and `\texteb{}`.

`{\sbseries\itshape Foo}` or `\textsb{\textit{Foo}}` print *Foo*.

**sfstext:** switches to *sans-serif* text fonts.

**fakedscripts:** prevents `realscripts` to be loaded, super- and sub-scripts will be produced by scaling (this is the way `pdflatex` works) which gives access to all characters available in the font but produces too tall and too thin glyphs. On the other hand `realscripts` gives access to the super/sub-scripts available in the font (a reduced set made of digits, lowercase latin letters, parentheses, etc. for the `Kpfonts`) which are better looking. I would recommend to keep `realscripts` loaded and to use the starred variants `\textsuperscript*`, `\textsubscript*`, to produce faked super/sub-scripts when needed, coding `H\textsuperscript*{\#}` to get  $H^{\#}$  while `H\textsuperscript{\#}` would output  $H^{\#}$  ( $\#$  missing as superscript).

**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’ and ‘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 ‘f’ unless it ends a word or is followed by *b*, *f* or *h*<sup>3</sup> and ancient ligatures *fi*, *fl*, *ft* are activated. Coding `\textit{some of Highlands’ mysterious castles...}` will print *some of Highlandf’ mysteriouf castlef...* which is correct.

The automatic substitution relies on the OpenType feature `StylisticSet=12`. A round ‘s’ (resp. long ‘f’) can be forced by coding `s=` or `\shorts{}` (resp. `f4` or `\longs{}`).

**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*.

---

3. Rules found on [wikipedia](#).

4. On Unix systems the Compose key can be used: Compose f s.

**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 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 maths fonts *only*

**lightmath**: uses *light* maths fonts.  
**sfmath**: uses *sans-serif* maths fonts.  
**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  $\mathbb{C}, \mathbb{K}, \mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{Z}$  (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 :  
 $\int\int\int$  and  $\int\int\int\int$  instead of  $\iiint$  and  $\iiiiint$  (default).  
**partialup**: the `\partial` symbol is printed upright  $\partial$  instead of  $\partial$ .  
**fancyReIm**: commands `\Re` and `\Im` print  $\Re$  and  $\Im$  respectively instead of  $\mathbb{R}$  and  $\mathbb{I}$ .  
**tight**: horizontal spaces tightened in maths 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>5</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.:

---

5. A careful 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.

```
\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 maths fonts, loading font spec and using the `\set...font` commands is enough, no need to load `kpfonts-otf` at all.

## 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>6</sup>, is **SmallCapsFeatures = {Letters=PetiteCaps}**.

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

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

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

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

**Ligatures=Rare** adds ct and 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='; this option should no longer be used, it is superseded by **StylisticSet=12** below.

**StylisticSet=12** has been described in option `veryoldstyle p. 3`: it operates a contextual substitution of round 's' by long 'f'. An alias **Style=autoLongs** is available if `kpfonts-otf.sty` has been loaded.

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

**StylisticSet=2**: `\SS` prints SS instead of  $\mathcal{B}$  (capital *Eszett*), ditto for small/petite caps.

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

Options may be are chosen for each font, say:

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

but can also be shared by different typefaces:

---

6. Changed in v0.37 to match the original `kpfonts` package.

```

\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`.

2. Note the + ending `\defaultfontfeatures+` : options are *added*, not overwriting any other (including those of `fontspec.cfg`).

3. Options can be gathered: `Ligatures={Rare,Historic}` (with braces) is equivalent to `Ligatures=Rare` and `Ligatures=Historic`.

4. These options can also be switched on and off *locally* using `\addfontfeatures` inside a group, f.i. to print lowercase digits 1234576890 with a font loaded with option `Numbers=Lining`:

```
{\addfontfeatures{Numbers=Lowercase}1234576890}
```

Actually, a shortcut is available in this case: `\oldstylenums{1234576890}`.

5. With the **KpRoman**, it is possible to define two more weights *Light* and *Semibold* borrowed from **KpLight**:

```

\newfontfamily\KpLight{KpLight}[<same options as KpRoman>]
\newcommand*\ltseries{\KpLight}
\newcommand*\sbseries{\KpLight\bfseries}
\DeclareTextFontCommand{\textlt}{\ltseries}
\DeclareTextFontCommand{\textsb}{\sbseries}

```

These weights are then available through `\ltseries`, `\sbseries` commands to be used in a group or alternatively through one argument commands `\textlt{}` and `\textsb{}`.

With the **KpLight**, weights *Semibold* and *Extrabold* can be defined similarly.

## 2.2 Options for maths fonts

The following options can be passed either to `unicode-math`<sup>7</sup> or to `\setmathfont{}`:

**math-style** = ISO, TeX (default), french, upright;

**bold-style** = ISO, TeX (default), upright;

**partial** = upright (default italic);

---

7. See the manual `unicode-math.pdf`.

`nabla` = italic (default upright);

Seven ‘Style Variants’ are available with Kp-fonts, here are the first three:

**Style=mathcal** (+ss01) commands `\mathcal{}` and `\mathscr{}` print  $ABC$  instead of  $\mathcal{ABC}$  (default), see note below;

**Style=bbsans** (+ss02) `\mathbb{}` prints Sans-Serif Blackboard bold capitals  $ABC$  for Serif maths fonts KpMath-Regular and KpMath-Light instead of  $\mathbb{ABC}$ ;

**Style=narrowiints** (+ss03) provides condensed repeated integrals:  $\iiint$  instead of  $\iiiii$  (default).

**Note:** if you want commands  $\mathcal{ABC}$  and  $\mathscr{ABC}$  to print  $ABC$  and  $\mathcal{ABC}$  respectively, you can use unicode-math’s option range this way:

```
\setmathfont{KpMath-Regular}[options]
```

```
\setmathfont{KpMath-Regular}[range={cal,bfcal},StylisticSet=1]
```

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>	$\not\leq$	$\not\leqslant$
<code>\ngeq</code>	$\not\geq$	$\not\geqslant$
<code>\leqq</code>	$\leq\leq$	$\leq\leqslant$
<code>\geqq</code>	$\geq\geq$	$\geq\geqslant$
<code>\leqless</code>	$\leq<$	$\leqslant<$
<code>\eqgtr</code>	$\leq>$	$\leqslant>$
<code>\lesseqgtr</code>	$\leq\approx$	$\leqslant\approx$
<code>\gtreqless</code>	$\approx\leq$	$\approx\leqslant$
<code>\lesseqqgtr</code>	$\leq\approx\approx$	$\leqslant\approx\approx$
<code>\gtreqqless</code>	$\approx\approx\leq$	$\approx\approx\leqslant$

Table 2 – Style=smaller (+ss05)

Command	Default	Variant
<code>\mid</code>	$ $	$\! $
<code>\nmid</code>	$\nmid$	$\! \nmid$
<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>	$\subsetneq$	$\subsetneqslant$
<code>\supsetneq</code>	$\supsetneq$	$\supsetneqslant$
<code>\subsetneqq</code>	$\subsetneqq$	$\subsetneqqslant$
<code>\supsetneqq</code>	$\supsetneqq$	$\supsetneqqslant$

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 shows the available ‘Glyphs Variants’:

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

Example: with `\setmathfont{KpMath-Regular.otf}[CharacterVariant={3,6}]` commands `\epsilon` and `\phi` print  $\epsilon$  and  $\phi$  instead of  $\epsilon$  and  $\phi$ . The same is true of course for all shapes 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 `\mihbar` (replacement for AMS’ command `\hbar`).

## 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 maths 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>	$\longmapsto$
<code>\mmapsfrom</code>	$\mapsfrom$		<code>\longmmapsfrom</code>	$\longmapsfrom$
<code>\Mmapsto</code>	$\Mmapsto$		<code>\Longmmapsto</code>	$\Longmmapsto$
<code>\Mmapsfrom</code>	$\Mmapsfrom$		<code>\Longmmapsfrom</code>	$\Longmapsfrom$
<code>\leftrighthdasharrow</code>	$\leftrightarrow$		<code>\leadsto</code>	$\leadsto$
<code>\boxright</code>	$\boxrightarrow$		<code>\boxleft</code>	$\boxleftarrow$
<code>\circcleright</code>	$\circlearrowright$		<code>\circleleft</code>	$\circlearrowleft$
<code>\Diamondright</code>	$\diamondrightarrow$		<code>\Diamondleft</code>	$\diamondleftarrow$
<code>\boxdotright</code>	$\boxdotrightarrow$		<code>\boxdotleft</code>	$\boxdotleftarrow$
<code>\circledotright</code>	$\circledotrightarrow$		<code>\circledotleft</code>	$\circledotleftarrow$
<code>\Diamonddotright</code>	$\diamonddotrightarrow$		<code>\Diamonddotleft</code>	$\diamonddotleftarrow$
<code>\boxRight</code>	$\boxRight$		<code>\boxLeft</code>	$\boxLeft$
<code>\boxdotRight</code>	$\boxdotRight$		<code>\boxdotLeft</code>	$\boxdotLeft$
<code>\DiamondRight</code>	$\DiamondRight$		<code>\DiamondLeft</code>	$\DiamondLeft$
<code>\DiamonddotRight</code>	$\DiamonddotRight$		<code>\DiamonddotLeft</code>	$\DiamonddotLeft$
<code>\multimapdot</code>	$\multimapdot$		<code>\multimapdotinv</code>	$\multimapdotinv$
<code>\multimapdotboth</code>	$\multimapdotboth$			
<code>\multimapbothvert</code>	$\multimapbothvert$		<code>\multimapdotbothvert</code>	$\multimapdotbothvert$
<code>\multimapdotbothAvert</code>	$\multimapdotbothAvert$		<code>\multimapdotbothBvert</code>	$\multimapdotbothBvert$
<code>\capplus</code>	$\capplus$		<code>\sqcupplus</code>	$\sqcupplus$
<code>\parallelslant</code>	$\parallel$		<code>\colonsim</code>	$\colonsim$
<code>\parallelbackslant</code>	$\parallel$		<code>\Colonsim</code>	$\Colonsim$
<code>\eqqColon</code>	$\equiv$		<code>\Colondash</code>	$\Colondash$
<code>\strictif</code>	$\rightarrow$		<code>\strictfi</code>	$\rightarrow$
<code>\circledvee</code>	$\vee$		<code>\circledwedge</code>	$\wedge$
<code>\openJoin</code>	$\times$		<code>\opentimes</code>	$\times$
<code>\lambdaslash</code>	$\lambda$		<code>\lambdabar</code>	$\lambda$
			<code>\strictiff</code>	$\rightarrow$
			<code>\circledbar</code>	$\circledbar$
			<code>\VvDash</code>	$\Vdash$
			<code>\Wr</code>	$\wr$

<code>\idotsint</code>	$\int \dots \int$	$\int \dots \int$		
<code>\ointclockwise</code>	$\oint$	$\oint$	<code>\varointctrlockwise</code>	$\oint$ $\oint$
<code>\oiintclockwise</code>	$\oiint$	$\oiint$	<code>\oiintctrlockwise</code>	$\oiint$ $\oiint$
<code>\varoiintclockwise</code>	$\varoiint$	$\varoiint$	<code>\varoiintctrlockwise</code>	$\varoiint$ $\varoiint$
<code>\oiintclockwise</code>	$\oiint$	$\oiint$	<code>\oiintctrlockwise</code>	$\oiint$ $\oiint$
<code>\varoiintclockwise</code>	$\varoiint$	$\varoiint$	<code>\varoiintctrlockwise</code>	$\varoiint$ $\varoiint$
<code>\sqiint</code>	$\sqiint$	$\sqiint$	<code>\sqiint</code>	$\sqiint$ $\sqiint$

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

### 3.3 Wide accents

— `\wideoverbar` and `\mathunderbar`<sup>8</sup>

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

— `\widehat` and `\widetilde`

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

— `\widecheck` and `\widebreve`

$$\check{x} \quad \widecheck{xxx} \quad \widecheck{xxxxx} \quad \breve{x} \quad \widebreve{xxx} \quad \widebreve{xxxxx}$$

— `\overparen` and `\underparen`

$$\overparen{x} \quad \overparen{xy} \quad \overparen{xyz} \quad \overparen{A \cup B} \quad \overparen{A \cup (B \cap C) \cup D} \quad \overparen{x+y} \quad \overparen{a+b+\dots+z}$$

$$\underparen{x} \quad \underparen{xz} \quad \underparen{xyz} \quad \underparen{x+z} \quad \underparen{a+b+\dots+z}$$

— `\overbrace` and `\underbrace`

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

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

8. `\overline` and `\underline` are not font related, they are based on `\rule`.

— `\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` (or `\overrightarrowarc`)

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

### 3.4 Maths 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>9</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:

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

`\mathversion{bold}`

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

`\mathversion{sans}`

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

To get bold formulas you do not need to define `\mathversion{bold}`, you can just use the `\boldmath` command: f.i. `\boldmath $E=mc^2$` prints  $E = mc^2$ .

---

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