

tikz-decofonts

Simple "decoration" fonts or effects,
made with TikZ, for small texts...

Version 0.20c - 2026/09/28

Cédric Pierquet

c pierquet - at - outlook . fr

<https://github.com/cpierquet/latex-packages/tree/main/tikz-decofonts>

```
%paintbrush effect
\tkzbrush[color=blue]{DECORATION}
%paintink effect
\tkzink[color=orange]{DECORATION}
%pixeletter effect
\tkzpixl[color=violet]{DECORATION}
%bicolor effect
\tkzbicolor[colors=blue/red,style=ndiag]{\Huge\sffamily DECORATION}
\tkzbicolor[colors=teal/orange,style=ellips]{\Huge\sffamily DECORATION}
%surround effect
\tkzsurround[color=orange]{$I= \displaystyle\int_a^b f(x) \text{trm}{d}x = F(b)-F(a)$}
```

DECORATION

DECORATION

Decoration

DECORATION DECORATION

$$I = \int_a^b f(x)dx = F(b) - F(a)$$

Ideas and parts of codes, with CC BY-SA 4.0 or LPPL 1.3c licenses, come from :

- [\[link on tex.stackexchange\]](#) from user121799 for paintbrush;
- [\[link on tex.stackexchange\]](#) from user121799 for inkbrush ;
- [\[link on github\]](#) from meyvenil0844-byte for multicancel.

Contents

1	Loading, useful packages, purposes and acknowledgments	3
1.1	Loading and useful packages	3
1.2	Purposes and acknowledgments	3
2	The different effects	4
2.1	Paint brush	4
2.2	Ink brush	5
2.3	Pixl brush	6
2.4	Bicolor effect	7
2.5	Comic bubble	9
2.6	Surround effect	10
2.7	Underline effect	10
2.8	Arrow with fitted text	11
2.9	Freehand circled text	13
2.10	Block (3D) of letters	15
2.11	Underline with pen pictogram	16
2.12	Calligraphy pen (like pencircle in MetaPost) with calligraphy library	17
2.12.1	Global usage	17
2.12.2	Predefined pens	19
2.13	Multiple cancellation lines	20
2.13.1	Credits	20
2.13.2	Usage	20
3	History	22

1 Loading, useful packages, purposes and acknowledgments

1.1 Loading and useful packages

In order to load `tikz-decofonts`, simply use:

```
\usepackage{tikz-decofonts}
```

Loaded packages are `tikz`, `tikz-3dplot`, `settoebox`, `xintexpr`, `simplekv`, `listofitems`, `xstring`, `ifthen` and `calc`.

Loaded `tikz` libraries are `decorations`, `decorations.pathreplacing`, `calc`, `positioning`, `bbox`, `calligraphy`, `arrows.meta`, `shapes.callouts` and `bending`.

Globally, height of rendering is about 1cm, but some customization are available.

1.2 Purposes and acknowledgments

The purposes of this small package is not to provide full fonts, with all characters, but give restricted possibility to present **short** texts of title with *cute* effects.

These decorations are not destined to be use for whole paragraphs or texts, but just for fun for small titles or small texts.

Since many calculations are required internally, compilation time can be increased, so be careful with intensive use!

Ideas and parts of codes, with CC BY-SA 4.0 licenses, come from:

- [\[link on tex.stackexchange\]](#) from user121799 for paintbrush;
- [\[link on tex.stackexchange\]](#) from user121799 for inkbrush ;
- [\[link on github\]](#) from meyvenil0844-byte for multicancel.

Some adjustments (keys, paths) are maid from original code, but the global usage is due to the links above!

2 The different effects

2.1 Paint brush

☛ The effect is **very** *time-consuming* to compile!!

```
%only uppercase letters allowed  
\tkzbrush[color=...,lines=...,scale=...]<tikzpicture options>{short text}  
  
%color = black by default  
%lines = 12 by default  
%scale = 1 by default
```

```
%default output  
\tkzbrush{TIKZ DECORATION}
```

TIKZ DECORATION

```
%customization(s) (external file for speeding rendering ^^)  
\tkzbrush[color=red,lines=20,scale=1.5]{CUSTOM TEXT}
```

CUSTOM TEXT

2.2 Ink brush

```
%only uppercase letters allowed  
\tkzink[color=...,thick=...,scale=...]<tikzpicture options>{short text}  
  
%color = black by default  
%thick = 3 by default  
%scale = 1 by default
```

```
%default output  
\tkzink{TIKZ DECORATION}
```

TIKZ DECORATION

```
%customization(s)  
\tkzink[color=olive,thick=5,scale=2]{TIKZ DECORATION}
```

TIKZ DECORATION

2.3 Pixl brush

```
%regular letters allowed (upper and/or lower), rendered characters are 5 columns x 7 lines
\tkzpixl%
```

```
[height=...,thick=...,color=...,gridcolor=...,border=...,
offseth=...,offsetv=...,gridafter=...,nospaceafter=...]
<tikzpicture options>{short text}
```

```
\tkzpixletter%
```

```
[height=...,thick=...,color=...,gridcolor=...,border=...,
offseth=...,offsetv=...,gridafter=...,nospaceafter=...]
<tikzpicture options>{letter}
```

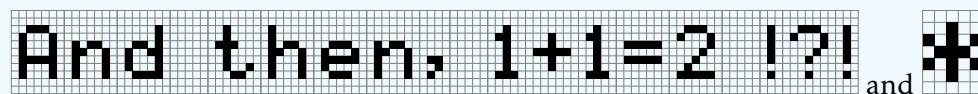
```
\tkzpixlquote%if problem with '...
```

```
[height=...,thick=...,color=...,gridcolor=...,
offseth=...,offsetv=...,gridafter=...,nospaceafter=...]
<tikzpicture options>
```

```
%height = 11mm by default, thick = 0.1mm by default
%color = black by default, gridcolor = gray by default
%offseth = 1 by default (squares left/right of regular character)
%offsetv = 2 by default (squares above/below of regular character)
%boder = false by default (boolean for adding border)
%gridafter = false by default (boolean for showing grid after pixls)
%nospaceafter = false by default (boolean for removeing hspace after pixls)
```

```
%default output, comparaison with character w/o offsets (5*7)
```

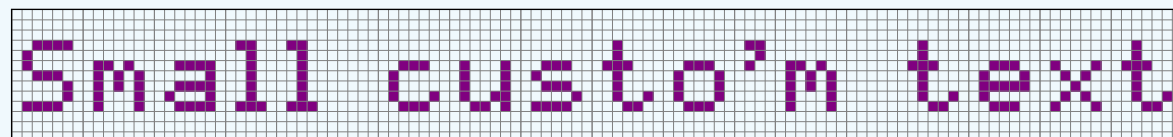
```
\tkzpixl{And then, 1+1=2 !?!} and \tkzpixletter[offseth=0,offsetv=0]{*}
```



The image shows two side-by-side outputs on a grid. The left output, generated by `\tkzpixl`, shows the text "And then, 1+1=2 !?!" where each character is centered within a 5x7 grid cell. The right output, generated by `\tkzpixletter` with `offseth=0, offsetv=0`, shows an asterisk character that is not centered but aligned to the top-left corner of its grid cell. The word "and" is written in a standard font between the two grid outputs.

```
%customization(s)
```

```
\tkzpixl[color=violet,height=1.75cm,gridafter,offsetv=3,border]{Small custo'm text}
```



The image shows the output of the `\tkzpixl` command with customization options. The text "Small custo'm text" is rendered in a violet color. The characters are larger, corresponding to a height of 1.75cm. The text is rendered on a grid, and the customization options include `gridafter`, `offsetv=3`, and `border`.

2.4 Bicolor effect

```
%compatible with all fonts and/or sizes
\tkzbicolor%
  [colors=...,style=...,intdelta=...]
  {text}

\tkzbicolor%with paral style
  [paralcolors=...,style=paral,intdelta=...]
  {text}

%style = midh/midv/rect/ellips/paral (paral by default)
%colors = red/blue by default,
%paralcolors = red/blue/orange/gray by default
%intdelta = 0.66 by default, percentage of int dim for ellips/rect
```

```
\def\sampleformula{\$\displaystyle\int_0^1 \dfrac{1+x}{1+x^2}\,,\,\mathrm{d}x\$}
\def\sampletxt{But I must explain to you how all this mistaken idea of denouncing pleasure.}
\def\samplenum{1500}
```

```
\tkzbicolor[style=midh]{\sampleformula}
or \tkzbicolor[colors=violet/magenta,style=midv]{\sampleformula}
or \tkzbicolor[colors=darkgray/olive,style=ndiag]{\sampleformula}
or \tkzbicolor[colors=orange/teal,style=sdiag]{\sampleformula}
or \tkzbicolor[colors=orange/teal,style=ellips]{\sampleformula}
or \tkzbicolor[colors=pink/violet,style=rect]{\sampleformula}
```

$$\int_0^1 \frac{1+x}{1+x^2} dx \text{ or } \int_0^1 \frac{1+x}{1+x^2} dx \text{ or } \int_0^1 \frac{1+x}{1+x^2} dx \text{ or } \int_0^1 \frac{1+x}{1+x^2} dx \text{ or } \int_0^1 \frac{1+x}{1+x^2} dx \text{ or } \int_0^1 \frac{1+x}{1+x^2} dx$$

```
\tkzbicolor[style=midh]{\sampletxt}\par
\tkzbicolor[colors=violet/magenta,style=midv]{\sampletxt}\par
\tkzbicolor[colors=darkgray/olive,style=ndiag]{\sampletxt}\par
\tkzbicolor[colors=orange/teal,style=sdiag]{\sampletxt}\par
\tkzbicolor[colors=orange/teal,style=ellips]{\sampletxt}\par
\tkzbicolor[colors=pink/violet,style=rect]{\sampletxt}
```

But I must explain to you how all this mistaken idea of denouncing pleasure.

But I must explain to you how all this mistaken idea of denouncing pleasure.

But I must explain to you how all this mistaken idea of denouncing pleasure.

But I must explain to you how all this mistaken idea of denouncing pleasure.

But I must explain to you how all this mistaken idea of denouncing pleasure.

But I must explain to you how all this mistaken idea of denouncing pleasure.

```

\tkzbicolor[style=midh]{\parbox{2.5cm}{\sampletxt}}
\tkzbicolor[colors=violet/gray,style=midv]{\parbox{2.5cm}{\sampletxt}}
\tkzbicolor[colors=darkgray/olive,style=ndiag]{\parbox{2.5cm}{\sampletxt}}
\tkzbicolor[colors=orange/cyan,style=sdiag]{\parbox{2.5cm}{\sampletxt}}
\tkzbicolor[colors=pink/violet,style=ellips]{\parbox{2.5cm}{\sampletxt}}
\tkzbicolor[colors=pink/violet,style=ellips,intdelta=0.85]{\parbox{2.5cm}{\sampletxt}}
\tkzbicolor[colors=cyan/darkgray,style=rect]{\parbox{2.5cm}{\sampletxt}}

```

But I must ex- But I must ex- But I must ex- But I must ex- But I must ex- But I must ex-
 plain to you plain to you plain to you plain to you plain to you plain to you
 how all this how all this how all this how all this how all this how all this
 mistaken idea mistaken idea mistaken idea mistaken idea mistaken idea mistaken idea
 of denouncing of denouncing of denouncing of denouncing of denouncing of denouncing
 pleasure. pleasure. pleasure. pleasure. pleasure. pleasure.
 But I must ex-
 plain to you
 how all this
 mistaken idea
 of denouncing
 pleasure.

```

\tkzbicolor[style=paral]{\parbox{6cm}{\large\sffamily\sampletxt}}

```

But I must explain to you how all
 this mistaken idea of denounc-
 ing pleasure.

```

\tkzbicolor[style=midh]{\samplenum}\par
\tkzbicolor[colors=violet/magenta,style=midv]{\samplenum}\par
\tkzbicolor[colors=darkgray/olive,style=ndiag]{\samplenum}\par
\tkzbicolor[colors=orange/teal,style=sdiag]{\samplenum}\par
\tkzbicolor[colors=orange/teal,style=ellips]{\samplenum}

```

1500
 1500
 1500
 1500
 1500

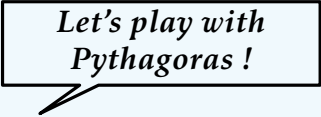
2.5 Comic bubble

```
%style tikz
\tikzset{comicbubble/.style args={#1#2}{%
  inner sep=1mm,line join=round,rectangle callout,draw,very thick,
  text width=#1,align=flush center,callout relative pointer=#2}}
```

```
%independant macro
\tkzcomicbubble[%
  width=...,coltxt=...,colbg=...,colframe=...,pospointer=(...),
  font=...,rcorners=true/false]%
<tikz options>{text}

%with tikzpicture environment
\begin{tikzpicture}
  %...
  \tkzcomicbubble*[%
    width=...,coltxt=...,colbg=...,colframe=...,pospointer=(...),
    font=...,rcorners=true/false]%
    <tikz options>{text}
\end{tikzpicture}
```

```
\def\mytext{Let's play with Pythagoras !\relax}
\tkzcomicbubble[font=\large\bfseries\itshape]{\mytext}~~
\tkzcomicbubble%
  [width=3cm,coltxt=red,colframe=teal,font=\bfseries,
  colbg=yellow!15,pospointer={(150:1em)},rcorners]%
  <thick>%
  {\mytext}
```

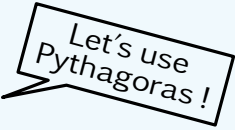


*Let's play with
Pythagoras !*



**Let's play with
Pythagoras !**

```
%\usepackage{setspace}
\def\mytext{Let's use Pythagoras !\relax}
{\tikz[rotate=-15,transform
  shape]\tkzcomicbubble*[width=2.5cm,font=\large\sffamily]<execute at begin
  node={\setlength{\baselineskip}{0.75\baselineskip}}>{\mytext};}
```



Let's use
Pythagoras !

2.6 Surround effect

```
\tkzsurround[width=...,node=...,color=...]{text}
```

%width = width of line (1.25pt by default)

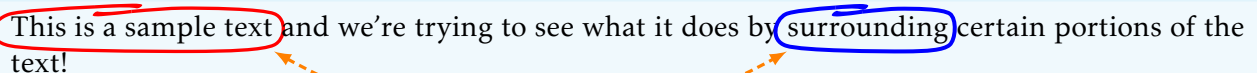
%color = red by default

%node = name of node (AAAAZ by default)

```
\tkzsurround[node=AAA]{This is a sample text} and we're trying to see what it does  
by \tkzsurround[color=blue,width=1.5pt,node=BBB]{surrounding} certain portions of the text!
```

```
{\tikz[remember picture,overlay]  
  \draw[densely dashed,orange,<->,very thick,>=latex]  
    ([yshift=-1mm]AAA.south east) to[bend right] ([yshift=-1mm]BBB.south west) ;  
}
```

This is a sample text and we're trying to see what it does by surrounding certain portions of the text!



2.7 Underline effect

```
\tkzunderline[width=...,node=...,color=...,height=...]{text}
```

%width = width of line (1.25pt by default)

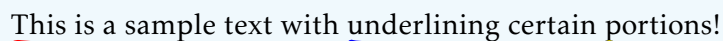
%color = red by default

%height = global height of underlining (10mm by default)

%node = name of node (AAAAZ by default)

```
\tkzunderline{This is a sample text} with  
\tkzunderline[color=blue,width=1.5pt,height=15mm]{underlining} certain  
\tkzunderline[color=yellow,width=1pt,height=15pt]{portions}!
```

This is a sample text with underlining certain portions!



2.8 Arrow with fitted text

```
\tkzbrush[<keys>]<tikzpicture options>{text}

%color      = color of arrow (gray)
%txtcolor   = color of text (white)
%width      = width 'with or without ending' width of arrow (2cm)
%bheight    = big height (8mm)
%sheight    = small height (4mm)
%hoffset    = text h offset (0.2mm)
%shadow     = boolean for shadow (true)
%shownodes  = boolean for helping nodes (false)
%expand     = use 'max' width for txt (false)
%globalwidth = boolean for width including 'ending' (false)
```

Nodes (eg for positionning) are:

- arrow-bottomleft
- arrow-topleft
- arrow-begin
- arrow-end
- arrow-textcenter
- arrow-bottom
- arrow-textright
- arrow-top
- arrow-smallbottom
- arrow-smalltop

```
\tkzfittextinarrow%
[bheight=3cm,sheight=1.75cm,width=6cm,color=orange,shownodes]%
{\vphantom{q}Demo}
```



```
\tkzfittextinarrow%
[bheight=3cm,sheight=1.75cm,width=6cm,color=teal,globalwidth,expand,txtcolor=yellow]%
{\vphantom{q}\bfseries\sffamily Demo}
```



```
\tkzfittextinarrow%
  [bheight=2cm,sheight=1cm,width=5cm,color=orange,expand]%
  {\vphantom{q}Démonstration}
```



Text before

```
\tkzfittextinarrow%
  [bheight=1em,sheight=0.6em,width=3em,color=green!50!black]%
  <rotate=-15,baseline=(arrow-smallbottom)>%
  {\itshape My examples}
\tkzfittextinarrow%
  [bheight=1em,sheight=0.6em,width=3em,color=green!50!black,expand]%
  <rotate=15,baseline=(arrow-textcenter)>%
  {\ttfamily My examples}
```

with text after

Text before   with text after

2.9 Freehand circled text

```
\tkzcircledtxt[<keys>]<tikzpicture options>{text}
```

%height	= height for manual insertion (2em)
%width	= width for manual insertion (2.3em)
%thickness	= thickness, percentage of height (0.05)
%absolute thickness	= absolute thickness, length (-)
%fill color	= bg color (lightgray!30)
%rule color	= boxrule color (black)
%rotate	= optional rotation (0)
%auto	= auto width/height/raise relative to len of inside text (0)
%fill	= boolean for filling (true)
%font	= optional font
%vstretch	= coefficient for vertical stretch, with auto usage (1.5)
%hoffset	= coefficient for horizontal offset with auto usage (0.8)
%random	= boolean for random effect (false)
%arabic	= boolean for arabic version of counter (false)
%alph	= boolean for alph version of counter (false)
%Alph	= boolean for Alpha version of counter (false)

%manual insertion (height + width + font)

```
{q\LARGE\sffamily\bfseries\tkzcircledtxt{99}\tkzcircledtxt[absolute thickness=1pt]{99}H}\\
{q\LARGE\sffamily\bfseries\tkzcircledtxt[random]{99}H}\\
{q\LARGE\sffamily\bfseries\tkzcircledtxt[random]{text}H}
```

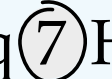
q   H

q  H

q  H

%automatic insertion relative to len of small text

```
{\Huge q\tkzcircledtxt[auto=1]{7}H or q\tkzcircledtxt[auto=2]{7}H}\\
{\Huge q\tkzcircledtxt[auto=2]{99}H or q\tkzcircledtxt[auto=3]{99}H}\\
{\Huge q\tkzcircledtxt[auto=2]{154}H or q\tkzcircledtxt[auto=3]{154}H }
```

q  H or q  H

q  H or q  H

q  H or q  H

%with custom font

```
{\Huge\sffamily q\tkzcircledtxt[font=\small\ttfamily,auto=3]{105}H}
```

q  H

`\tkzcircledtxt[auto=1]{7} Let  $f(x)=x^2+5x-3$ .`

⑦ Let f defined by $f(x) = x^2 + 5x - 3$.

```
%with enumitem package
\begin{enumerate}[label={%
  \color{darkgray}%
  \tkzcircledtxt[thickness=0.075,fill=false,arabic,font=\large\bfseries,auto=1,%
    fill color=teal!25,rule color=orange]{\arabic*}%
  }%
]
\item AAF
\item AAF
\begin{enumerate}[label={%
  \color{red}%
  \tkzcircledtxt[thickness=0.0375,fill=false,offset=0.25,alph,%
    font=\bfseries,auto=1,rule color=teal]{\alph*}%
  }%
]
\item AA
\item BGHMG
\end{enumerate}
\end{enumerate}
\item AAF
\item AAF
\item AAF
\item AAF
\item AAF
\item AAF
\item AAF
\end{enumerate}
```

- ① AAF
- ② AAF
 - Ⓐ AA
 - Ⓑ BGHMG
- ③ AAF
- ④ AAF
- ⑤ AAF
- ⑥ AAF
- ⑦ AAF
- ⑧ AAF
- ⑨ AAF

2.10 Block (3D) of letters

```
\tkzcircledtxt[<keys><tikzpicture options>{list of blocks}
```

```
%height      = height dim or auto/dauto for inline usage (dauto)
%hstretch     = h sep coefficient (1.05)
%spacestretch = space sep coefficient (0.25)
%dstrut       = characters for dauto usage (qH)
%strut        = characters for auto usage (H)
%font         = font for labels (\huge\bfseries\sffamily)
```

```
%list of blocks within letter1/colorblock1/colorletter1,...
%use * for 'space'
```

```
{\LARGE%
q%
\tkzletterblock%
{%
  C/red/white,H/blue/white,A/violet/white,%
  P/yellow/black,T/white/black,E/orange/black,%
  R/teal/white,*/magenta/white,0/cyan/white,7/pink/white%
}%
H%
}
```

qCHAPTER07H

```
\sffamily\bfseries\Huge%
H%
\tkzletterblock%
[height=auto]%
{%
  L/red/white,A/blue/white,T/violet/white,E/cyan/white,X/pink/white%
}%
H
```

H L A T E X H

```
%green lines for visualizing manual height
{\tikz\draw[thick,green] (0,0)--(0,1.75cm);}
\tkzletterblock%
[height=1.75cm,hstretch=1.25,font=\Huge\ttfamily,spacestretch=0.5]%
{%
  D/yellow/black,E/white/black,*//,C/orange/black,0/teal/white%
}%
{\tikz\draw[thick,green] (0,0)--(0,1.75cm);}

```

D E C O

2.11 Underline with pen pictogram

```
%main macro
\tkzunderlinepen[keys]{text}
%macro with hphantom of pen
\phantomulnewithpen

%thick          = thickness (0.8pt)
%color          = color for underling
%chardepth      = characters for depth calculus
%vstretchoffset = float for raising depth (0.5)
%scale          = scale of pen (1)
%swap           = boolean for 'downside' version (false)
```

```
\tkzunderlinepen{Exercise n°1}\par
\lipsum[1][1-3]

\tkzunderlinepen[swap]{Exercise n°1}\par
\phantomulnewithpen\lipsum[1][1-3]
```

Exercise n°1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

Exercise n°1

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

```
{\bfseries\sffamily\LARGE\textcolor{blue}{\tkzunderlinepen[thick=1pt]{Exercise n°2}}}\par
\lipsum[1][1-3]

{\bfseries\sffamily\LARGE\textcolor{violet}{\tkzunderlinepen[color=teal]{Exercise n°3}}}\par
\lipsum[1][1-3]

{\bfseries\sffamily\LARGE\textcolor{brown}{\tkzunderlinepen[color=orange]{Exercise
  (important) n°4}}}\par
\lipsum[1][1-3]
```

Exercise n°2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

Exercise n°3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

Exercise (important) n°4

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

2.12 Calligraphy pen (like pencircle in MetaPost) with calligraphy library

2.12.1 Global usage

This macro is a wrapper around the TikZ `calligraphy` library, designed to mimic the `pencircle` syntax from METAPOST:

```
pickup pencircle xscaled penunita yscaled penunitb rotated angle ;
```

The `calligraphy` library models the pen as two orthogonal segments (major and minor axes), whose lengths control the stroke width variation.

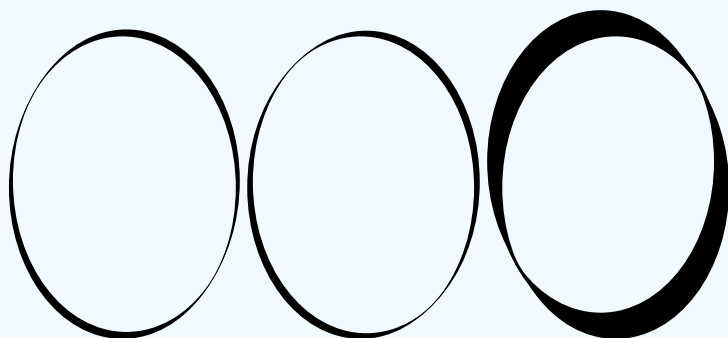
See the doc: <https://ctan.org/pkg/spath3> within Use for calligraphy.

```
%configure the pen (to use before \calligraphy)
\calligraphypencircle(*)[keys]

%starred (*) = outside tikz environment
%angle       = angle of the pen (60)
%line width  = width of the pen (3bp)
%ratio       = ratio xy axis (1/8)
%name        = name of defined pen (empty)
%double      = boolean for both axis (true)

%then use calligraphy as usual
\calligraphy[options] path ;
```

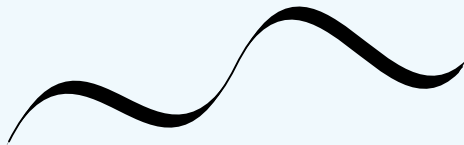
```
\begin{tikzpicture}
  \calligraphypencircle
  \calligraphy (0,0) circle[x radius=1.5cm,y radius=2cm] ;
\end{tikzpicture}
\begin{tikzpicture}
  \calligraphypencircle[angle=45]
  \calligraphy (0,0) circle[x radius=1.5cm,y radius=2cm] ;
\end{tikzpicture}
\begin{tikzpicture}
  \calligraphypencircle[angle=-60,line width=4mm,ratio=0.1]{}
  \calligraphy (0,0) circle[x radius=1.5cm,y radius=2cm] ;
\end{tikzpicture}
```



```

\begin{tikzpicture}
  \calligraphypencircle[line width=2mm]
  \calligraphy[]
  (0,0) .. controls (1,2) and (2,-1) .. (3,1)
  .. controls (4,3) and (5,0) .. (6,1) ;
\end{tikzpicture}

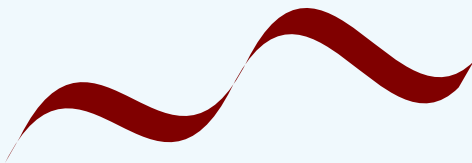
```



```

\begin{tikzpicture}
  \calligraphypencircle[line width=4mm,double=false]
  \calligraphy[pen colour=red!50!black]
  (0,0) .. controls (1,2) and (2,-1) .. (3,1)
  .. controls (4,3) and (5,0) .. (6,1) ;
\end{tikzpicture}

```

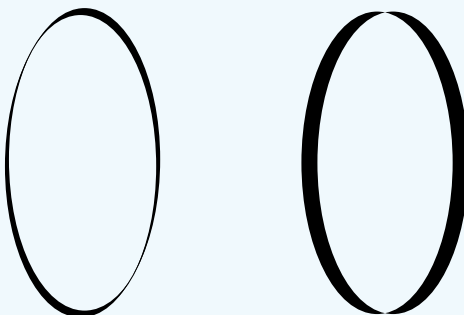


```

\calligraphypencircle*[name=styleA,angle=60,line width=3bp]
\calligraphypencircle*[name=styleB,angle=0, line width=6bp,double=false]

\begin{tikzpicture}
  \calligraphy[use pen=styleA] (0,0) circle[x radius=1cm, y radius=2cm] ;
  \calligraphy[use pen=styleB] (4,0) circle[x radius=1cm, y radius=2cm] ;
\end{tikzpicture}

```

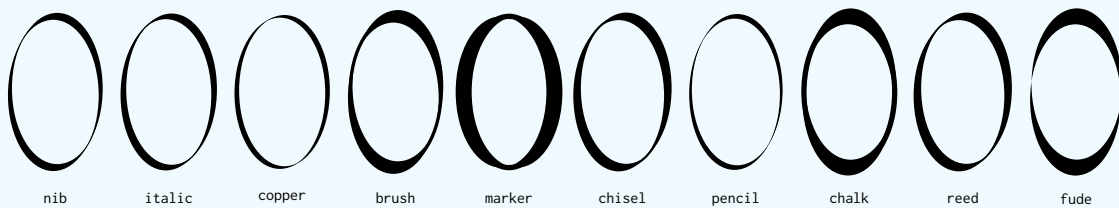


2.12.2 Predefined pens

The following pens are predefined and available without any configuration. Each pen is available in English and French.

%English / Français	-- angle	lw	ratio
%-----	%-----	%-----	%-----
%nib / plume	-- 60°	3bp	1/8
%italic / italique	-- 45°	3bp	1/6
%copper / cuivre	-- 30°	2bp	1/10
%brush / pinceau	-- 70°	5bp	1/4
%marker / marqueur	-- 0°	6bp	1/3
%chisel / ciseau	-- 45°	4bp	1/5
%pencil / crayon	-- 60°	2bp	1/12
%chalk / craie	-- 80°	6bp	1/3
%reed / roseau	-- 45°	4bp	1/6
%fude / japonais	-- 85°	6bp	---

```
\begin{tikzpicture}
  \foreach \penname/\xpos in {%
    nib/0,italic/1.5,copper/3,brush/4.5,marker/6,%
    chisel/7.5,pencil/9,chalk/10.5,reed/12,fude/13.5%
  }{%
    \begin{scope}[xshift=\xpos cm]
      \calligraphy[use pen=\penname]
      (0,0) circle[x radius=0.6cm, y radius=1cm] ;
      \node[below, font=\tiny\ttfamily] at (0,-1.2cm) {\penname} ;
    \end{scope}
  }
\end{tikzpicture}
```



```
\begin{tikzpicture}
  \calligraphy[use pen=fude, taper]
  (0,0) .. controls (2.5,1.5) and (-3.5,2.75) .. (0,3) ;
\end{tikzpicture}
```



2.13 Multiple cancellation lines

2.13.1 Credits

Original version of code come from meyvenil0844-byte user, within its [\[github\]](#) repository !

2.13.2 Usage

Parallel cancellation lines (*stacked* or *counting* style) over an expression, in maths or in text, with a chosen type, number of lines, color, or an arrow with a target value.

```
\tkzmulticancel<keys><tikzpicture options>{expression}

%type      = slash (default), backslash, cross or to (last line is an arrow)
%target     = text at the end of the arrow, for type=to (empty by default)
%lines      = number of lines (1 by default)
%color      = color of the lines (black by default)
%centered   = boolean for the 'counting' style, see below (false by default)
%sep        = gap between two lines (2.5pt by default) or two bars if centered (0pt)
%extend     = overshoot of the lines out of the expression (1.5pt by default)
%arrow      = arrow tip for type=to (-latex by default)

%TikZ style 'tkzmulticancel line' (semithick by default) is applied to all the lines
```

```
%default (stacked) style
$\tkzmulticancel{x+y}$ \quad
$\tkzmulticancel[lines=3,color=red]{x+y}$ \quad
$\tkzmulticancel[type=backslash,lines=2,color=blue]{A\cdot B}$ \quad
$\tkzmulticancel[type=cross,lines=3,color=purple]{M_n}$ \quad
$\tkzmulticancel[type=to,target=0,lines=2,color=orange]{f(x)-f(x)}$
```

$x+y$  $A \backslash B$  $f(x) \xrightarrow{0} f(x)$

Outside of math mode, the argument is typeset as text.

```
A \tkzmulticancel[lines=2,color=red]{cancelled} word.
```

A  word.

By default, the lines are **stacked**: they are parallel and shifted *perpendicularly* to their direction. With the boolean key `centered`, n bars are drawn side by side instead, like counting marks: they are evenly spread over the **whole width** of the expression and are as **tall** as it is, so they adapt to its size (with `lines=1`, both styles give the same diagonal).

```
%centered style
$\tkzmulticancel[lines=3,color=red,centered]{x+y}$ \quad
$\tkzmulticancel[type=backslash,lines=3,color=blue,centered]{A\cdot B}$ \quad
$\tkzmulticancel[type=cross,lines=2,color=purple,centered]{M_n}$ \quad
$\tkzmulticancel[type=to,target=0,lines=3,color=orange,centered]{f(x)-f(x)}$
```

$x+y$  $A \backslash B$  $f(x) \xrightarrow{0} f(x)$

Only a part of an equation can be cancelled, and the bars follow the size of the cancelled part:

```

 $\cancel{x}$ 
 $\frac{\cancel{(x+1)}}{\cancel{(x+1)}(x-2)}$ 
 $e^{\cancel{2x}}$ 

```

$$\cancel{x} \quad \frac{\cancel{(x+1)}}{\cancel{(x+1)}(x-2)} \quad e^{\cancel{2x}}$$

Gaps, overshoot and arrow tip can be customized with keys.

```

 $\cancel{x+y}$ 
 $\cancel{x \neq y}$ 
 $\cancel{a \rightarrow a^0}$ 

```

$$\cancel{x+y} \quad \cancel{x \neq y} \quad \cancel{a \rightarrow a^0}$$

The TikZ style `tkzmulticancel line` is applied to every line, so it can be appended (globally or in a group) to change the aspect of the lines.

```

\tikzset{tkzmulticancel line/.append style={dashed,line width=1pt}}
 $\cancel{x \neq y}$ 

```

$$\cancel{x \neq y}$$

The `<tikzpicture options>` are given to the `tikzpicture` containing the expression, which is a node named `content`. They can also be used to change the style of the lines for a single call, for example `<tkzmulticancel line/.append style={dashed}>`.

Each cancellation is a small `tikzpicture`, so the compilation time increases with intensive use!

3 History

0.20c: Multicancel
0.20b: Bugfix with AtBeginDocument
0.20a: Pencircle effect
0.1.8: Pen-underline
0.1.7: AlphabetCubes
0.1.6: Thickness for 'freehand' circled text
0.1.5: Small 'freehand' circled text
0.1.4: Arrow with fitted text
0.1.3: Underline or surround effect
0.1.2: Comic bubble
0.1.1: Bicolor effect
0.1.0: Initial version