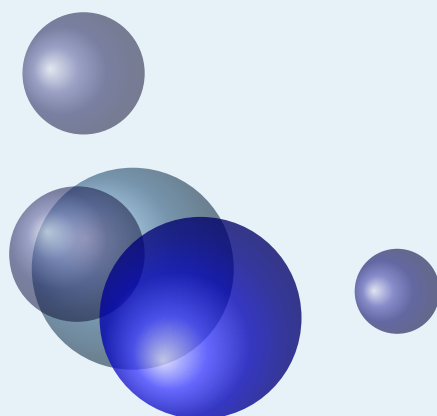


AlterMundus



Alain Matthes

October 20, 2026 Documentation V.1.51

<http://altermundus.fr>

tkzexample.sty

AlterMundus

Alain Matthes

`tkzexample.sty` is a package to present examples and their sources. It is essential if you want to compile the documentation of my packages like `tkz-euclide`. I needed for my first packages to work in UTF8 and to present my sources in French. These sources had accents so I had to adapt the code used by T. Tantau to present the examples of `pgfmanual`. There are still some imperfections so I invite you to improve this code if necessary

Please report typos or any other comments to this documentation to: [Alain Matthes](#).

This file can be redistributed and/or modified under the terms of the $\text{\LaTeX}$ Project Public License Distributed from [CTAN](#) archives.

Contents

1	How to use <code>tkzexample</code>	4
1.1	Loading the package	4
1.2	The two environments	4
1.3	How the result and the code are placed	4
1.4	Options of the environment <code>tkzexample</code>	4
1.5	Options of the environment <code>tkzltxexample</code>	6
1.6	Colors	6
1.7	Settings for the whole document	6
1.8	Saving the code in files	6
2	First example : no options	7
2.0.1	Minipage and accents without options	7
3	Inversion of the boxes: option <code>overhang</code>	7
4	Boxes with the same width: option <code>vbox</code>	9
5	Side by side: option <code>latex</code>	10
5.0.1	<code>latex=7cm</code>	10
5.0.2	<code>latex=8cm</code>	10
6	Side by side: option <code>width</code>	11
7	Option <code>code</code> and option <code>graphic</code>	12
8	Option <code>num</code>	13
8.1	Example 1	13
8.2	Example 2	13
9	Size option for code fonts	15
10	Option <code>code only</code>	16
11	Long code: option <code>break code</code>	17
12	Options of the frame of <code>code only</code> and <code>tkzltxexample</code>	19
13	Option <code>saved</code> of the package	19
14	History	19

1 How to use tkzexample

1.1 Loading the package

```
\usepackage{tkzexample}          % or
\usepackage[saved]{tkzexample}   % the code of each example is saved in a file
```

`tkzexample` loads `calc`, `xcolor`, `pgfkeys`, `mdframed` and `fancyvrb`. To execute some TikZ code, you have to load `tikz` (or `tkz-euclide`, etc.) yourself. The package works with `pdflatex`, `xelatex` and `lualatex`, and the code may contain accented letters (UTF-8).

1.2 The two environments

```
\begin{tkzexample}[options]      \begin{tkzltxexample}[options]
code                             code
\end{tkzexample}                 \end{tkzltxexample}
```

- `tkzexample`: the code is *executed* and *displayed*: the result and the source are placed side by side or one above the other.
- `tkzltxexample`: the code is only *displayed*, in a frame (as with the option `code only` of `tkzexample`).

The square brackets are optional. The text between `\begin{...}` and `\end{...}` is read verbatim: all the characters (`\` `{` `}` `$` `&` `#` `^` `_` `~` `%`) are displayed as they are written. The comments (`%` until the end of the line) are displayed in slanted font and are not executed.

1.3 How the result and the code are placed

1. The code is executed in a box. If the width of the result is less than the value of the option `width` (default 6cm), the result and the code are placed *side by side*: the result on the left, the code on the right (the option `overhang` swaps them). The width of the code is `\linewidth-1.5em-hsep-width`.
2. If the result is wider than `width`, the result is placed first and the code under it, on the whole width of the line.
3. With the option `code only`, the code is not executed; it is displayed in a frame (`mdframed`) that can be split across pages.
4. If the code is too long for the page, the option `break code` (section 11) places the result first and the code under it, in a frame that can be split across pages.

1.4 Options of the environment `tkzexample`

Layout

option	default	meaning
width=<dim> latex=<dim>	6cm	maximal width of the result for the side-by-side layout the code is executed in a minipage of width <dim> (for text); sets also <code>width</code>
vbox		the code is executed in a <code>\vbox</code> of the width of the line (for text, paragraphs)
overhang	false	the code first, then the result (boolean: <code>overhang=false</code>)
hsep=<dim>	0pt	horizontal space between the result and the code; with 0pt the space is <code>\hfill</code>
code only		the code is not executed, only displayed in a frame
execute code	true	boolean; <code>execute code=false</code> is like <code>code only</code>
break code=<v>	auto	auto, fill, always or never (section 11); <code>break code</code> alone means <code>fill</code>
above skip=<code>	<code>\par</code>	code inserted before the example, e.g. <code>above skip=\par\medskip</code>
below skip=<code>	<code>\par</code>	code inserted after the example

Colors and frames

option	default	meaning
code=<color>	white	background of the code
graphic=<color>	white	background of the result
frame code=<color>		frame around the code (<code>\fcolorbox</code>)
frame tex=<color>		frame around the result (<code>\fcolorbox</code>)

Text of the code

option	default	meaning
code style=<fonts> normal, small, very small	<code>\small\ttfamily</code>	font of the code size of the code (<code>\normalsize</code> , <code>\small</code> , <code>\footnotesize</code>)
num		numbers the lines (from 1)
global num		numbers the lines, the numbering continues from the previous example
numcolor=<color>	gray	color of the numbers
numbkgcolor=<color>	white	background of the numbers
pre=<code>, post=<code>		code executed before / after the code of the example (not displayed)

Frame of the code with code only (options of mdframed)

option	default	meaning
left margin, right margin	0pt	outer margins of the frame
inner left margin	0pt	
inner right margin	10pt	
inner top margin	6pt	
inner bottom margin	.4\baselineskip	
line frame width	0pt	width of the line of the frame

1.5 Options of the environment tkzltxexample

The following options of `tkzexample` are available: `code` (background of the frame), `num`, `global num`, `numcolor`, `numbkgcolor`, `code style`, `normal`, `small`, `very small` and the options of the frame (`left margin`, ..., `line frame width`). The option `leave comments` displays the comments as they are written (not in slanted font).

1.6 Colors

The colors can also be defined for the whole document with `\colorlet`:

color	default	used for
<code>graphicbackground</code>	white	background of the result (option <code>graphic</code>)
<code>codebackground</code>	white	background of the code (option <code>code</code>)
<code>codeonlybackground</code>	white	background of the frame of <code>code only</code> and <code>tkzltxexample</code>
<code>numbackground</code>	white	background of the line numbers
<code>numcolor</code>	gray	line numbers
<code>textcodecolor</code>	black	text of the code

1.7 Settings for the whole document

The style `every tkzexample` is used by each example; its default value is `width=6cm`. To add some options to all the examples:

```
\pgfkeys{/tkzexample/every tkzexample/.append style={code=gray!10,break code}}
```

With `.style` instead of `.append style`, the default value `width=6cm` is replaced. For `tkzltxexample`, the style is `/tkzltxexample/every tkzltxexample`.

1.8 Saving the code in files

With the package option `saved`, the code of each example is written in a file and then read (section 13). The name of the file is:

```
tkzFile-<section>-<subsection>-<subsubsection>.tex
```

The command `\tkzSavedCode[prefix=<name>]` changes the environment `tkzexample`: the code is only written in a file, without any display.

2 First example : no options

`tkzexample.sty` is based on the code of `codeexample` of T.Tantau . I wanted to be able to use the accents of the French language in the sources.

How to use this environment? Without any options, the interpretation of a code and the source of this code are placed one above the other.

To do this, simply place the source code in a `tkzexample`. The colors of the different boxes show their sizes. The colors are obtained as follows:

```
\colorlet{graphicbackground}{blue!10!white}
\colorlet{codebackground}{blue!20!lightgray!20}
```

2.0.1 Minipage and accents without options

This is a test for "minipage" and accents èçéà&\$. Since version 1.50 the brackets are optional.

```
\begin{minipage}{8cm}
\subsubsection{Minipage and accents without options}
This is a test for "minipage" and accents èçéà\&$ . Since version 1.50 the brackets are optional.
\end{minipage}
```

```
\begin{tkzexample}[]
\begin{minipage}{8cm}
\subsubsection{Minipage and accents without options}
This is a test for "minipage" and accents èçéà\&$ . Since version 1.50 the brackets are optional.
\end{minipage}
\end{tkzexample}
```

Since version 1.50, the square brackets are optional. These two forms are valid:

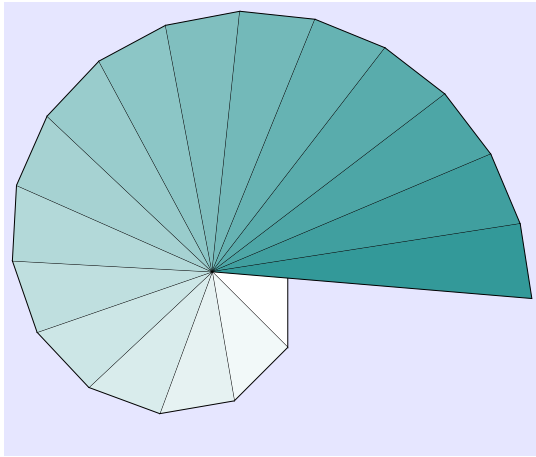
```
\begin{tkzexample}
\begin{tkzexample}[]
```

3 Inversion of the boxes: option overhang

`overhang` allows to swap the source and the result. `overhang` is a boolean: `overhang=false` gives the default placement (useful if `overhang` is set globally).

```
\begin{tkzexample}[overhang]
\begin{tikzpicture}
...
\end{tkzexample}
```

```
\begin{tikzpicture}
\tkzDefPoint(0,0){O}
\tkzDefPoint(1,0){a0}
\tkzDrawSegment(O,a0)
\foreach \i [count=\j] in {0,...,16}{%
\tkzDefPointWith[orthogonal normed](a\i,0)
\tkzGetPoint{a\j}
\pgfmathsetmacro{\c}{5*\i}
\tkzDrawPolySeg[fill=teal!\c](a\i,a\j,0)}
\end{tikzpicture}
```

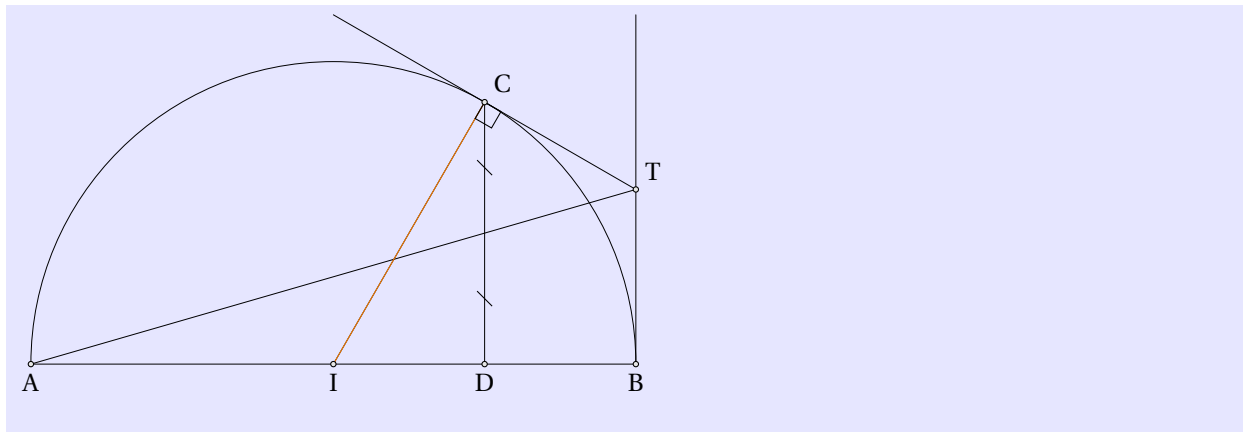


4 Boxes with the same width: option `vbox`

In order to present the code under the interpretation, I set the vbox option so that the interpretation and the source are in a box of the same maximum width.

```
\begin{tkzexample}[vbox]
    ...
\end{tkzexample}
```

which gives



```

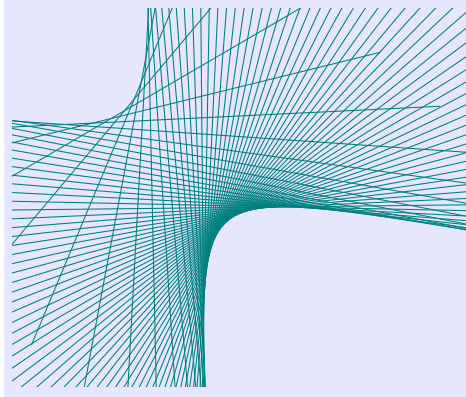
\begin{tikzpicture}[scale=1]
  \tikzset{new/.style={color=orange,line width=.2pt}}
  \tkzDefPoint(0,0){A}\tkzDefPoint(6,0){D}
  \tkzDefPoint(8,0){B}\tkzDefPoint(4,0){I}
  \tkzDefLine[orthogonal=through D](A,D)
  \tkzInterLC[R](D,\tkzPointResult)(I,4)\tkzGetSecondPoint{C}
  \tkzDefLine[orthogonal=through C](I,C)\tkzGetPoint{c}
  \tkzDefLine[orthogonal=through B](A,B)\tkzGetPoint{b}
  \tkzInterLL(C,c)(B,b)\tkzGetPoint{T}
  \tkzInterLL(A,T)(C,D)\tkzGetPoint{P}
  \tkzDrawArc(I,B)(A)
  \tkzDrawSegments(A,B A,T C,D I,C)\tkzDrawSegment[new](I,C)
  \tkzDrawLine[add = 1 and 0](C,T)\tkzDrawLine[add = 0 and 1](B,T)
  \tkzMarkRightAngle(I,C,T)
  \tkzDrawPoints(A,B,I,D,C,T)
  \tkzLabelPoints(A,B,I,D)\tkzLabelPoints[above right](C,T)
  \tkzMarkSegment[pos=.25,mark=s|](C,D)\tkzMarkSegment[pos=.75,mark=s|](C,D)
\end{tikzpicture}

```

5 Side by side: option latex

You must indicate the width of the box assigned to the interpretation, this is done with the option `latex`:

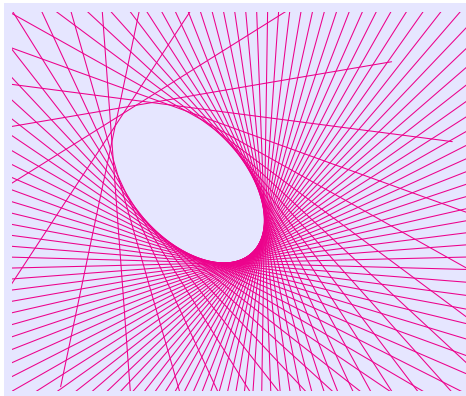
5.0.1 latex=7cm



```
\begin{tikzpicture}[scale=.5]
\tkzInit[xmin=-6,ymin=-4,xmax=6,ymax=6]
\tkzClip
\tkzSetUpLine[thin,color=teal]
\tkzDefPoint(0,0){O}
\tkzDefPoint(132:5){A}
\tkzDefPoint(4,0){B}
\foreach \ang in {5,10,...,360}{%
\tkzDefPoint(\ang:4){M}
\tkzDefLine[mediator](A,M)
\tkzGetPoints{x}{y}
\tkzDrawLine[add= 3 and 3](x,y)}
\end{tikzpicture}
```

```
\begin{tkzexample}[latex=7cm]
...
\end{tkzexample}
```

5.0.2 latex=8cm



```
\begin{tikzpicture}[scale=.5]
\tkzInit[xmin=-6,ymin=-4,xmax=6,ymax=6]
\tkzClip
\tkzSetUpLine[thin,color=magenta]
\tkzDefPoint(0,0){O}
\tkzDefPoint(132:4){A}
\tkzDefPoint(5,0){B}
\foreach \ang in {5,10,...,360}{%
\tkzDefPoint(\ang:5){M}
\tkzDefLine[mediator](A,M)
\tkzGetPoints{x}{y}
\tkzDrawLine[add= 3 and 3](x,y)}
\end{tikzpicture}
```

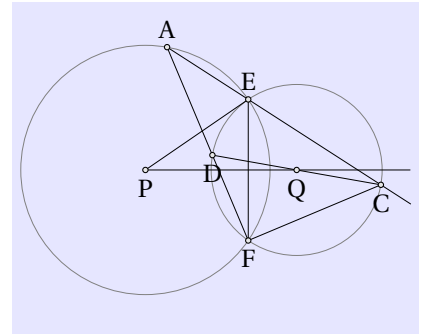
```
\begin{tkzexample}[latex=8cm]
...
\end{tkzexample}
```

6 Side by side: option width

If the option `latex= x cm` allows to reserve a box of width x cm for the code, the option `width= x cm` allows to reserve a box of width x cm for the result.

We leave a box of 8cm for the result or drawing and the rest of the line will contain a minipage of 8cm. As you can see a new section has been created and the code is contained in a 8cm box

```
\begin{tikzpicture}[scale=.4]
\tkzDefPoints{0/0/P,5/0/Q,3/2/I}
\tkzDefCircle[orthogonal from=P](Q,I)
\tkzGetFirstPoint{E}
\tkzDrawCircles(P,E Q,E)
\tkzInterCC[common=E](P,E)(Q,E) \tkzGetFirstPoint{F}
\tkzDefPointOnCircle[through = center P angle 80 point E]
\tkzGetPoint{A}
\tkzInterLC[common=E](A,E)(Q,E) \tkzGetFirstPoint{C}
\tkzInterLL(A,F)(C,Q) \tkzGetPoint{D}
\tkzDrawLines[add=0 and .75](P,Q)
\tkzDrawLines[add=0 and 2](A,E)
\tkzDrawSegments(P,E E,F F,C A,F C,D)
\tkzDrawPoints(P,Q,E,F,A,C,D)
\tkzLabelPoints(P,Q,F,C,D)
\tkzLabelPoints[above](E,A)
\end{tikzpicture}
```



The options used to get this result are :

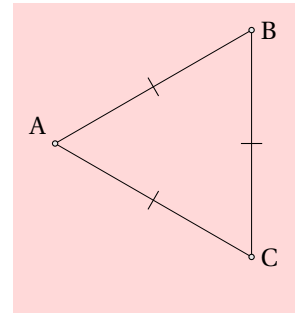
```
\begin{tkzexample}[overhang,width=6cm,code=red!10]
...
\end{tkzexample}
```

7 Option code and option graphic

This option is used to color the background of the code.

```
\begin{tkzexample}[overhang,width=9cm,code=gray!20,graphic=red!15]
...
\end{tkzexample}
```

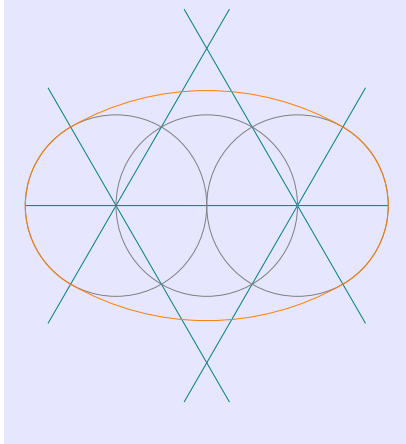
```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(2,3){A}
\tkzDefShiftPoint[A](30:3){B}
\tkzDefShiftPoint[A](-30:3){C}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints[right](B,C)
\tkzLabelPoints[above left](A)
\tkzMarkSegments[mark=|](A,B A,C B,C)
\end{tikzpicture}
```



8 Option num

You can number the lines of code. `numcolor` and `numbgcolor` are used to define the style of the numbering.

8.1 Example 1



```

1 \begin{tikzpicture}[scale=0.3]
2   \tkzDefPoint(-4,0){I}
3   \tkzDefPoint(4,0){J}
4   \tkzDefPoint(0,0){O}
5   \tkzInterCC(J,O)(O,J) \tkzGetPoints{L}{H}
6   \tkzInterCC(I,O)(O,I) \tkzGetPoints{K}{G}
7   \tkzInterLL(I,K)(J,H) \tkzGetPoint{M}
8   \tkzInterLL(I,G)(J,L) \tkzGetPoint{N}
9   \tkzDefPointsBy[symmetry=center J](L,H){D,E}
10  \tkzDefPointsBy[symmetry=center I](G,K){C,F}
11  \begin{scope}[line style/.style = {very thin,teal}]
12    \tkzDrawLines[add=1.5 and 1.5](I,K I,G J,H J,L)
13    \tkzDrawLines[add=.5 and .5](I,J)
14    \tkzDrawCircles(O,I I,O J,O)
15    \tkzDrawArc[delta=0,orange](N,D)(C)
16    \tkzDrawArc[delta=0,orange](M,F)(E)
17    \tkzDrawArc[delta=0,orange](J,E)(D)
18    \tkzDrawArc[delta=0,orange](I,C)(F)
19  \end{scope}
20 \end{tikzpicture}

```

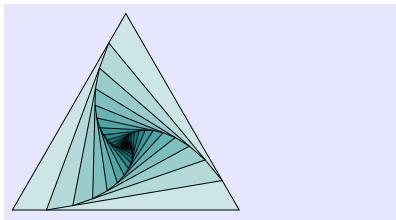
this is obtained with:

```

\begin{tkzexample}[width=6cm,num,numcolor=red,numbgcolor=white,code style=\ttfamily]
...
\end{tkzexample}

```

8.2 Example 2



```

1 \begin{tikzpicture}[scale=.25]
2   \tkzDefPoints{0/0/A,12/0/B,6/12*sind(60)/C}
3   \foreach \density in {20,30,...,240}{%
4     \tkzDrawPolygon[fill=teal!\density](A,B,C)
5     \pgfnodealias{X}{A}
6     \tkzDefPointWith[linear,K=.15](A,B) \tkzGetPoint{A}
7     \tkzDefPointWith[linear,K=.15](B,C) \tkzGetPoint{B}
8     \tkzDefPointWith[linear,K=.15](C,X) \tkzGetPoint{C}
9   }
\end{tikzpicture}

```

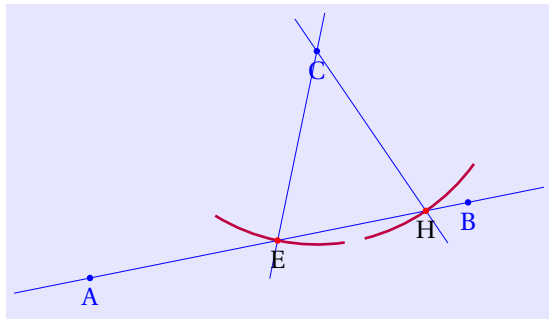
```

\begin{tkzexample}[latex=5cm,code=red!20,num,small]
...
\end{tkzexample}

```

The global num option allows to have a global numbering of the lines with the next code:

```
\begin{tkzexample}[latex=7cm,global num,small,hsep=24pt]
...
\end{tkzexample}
```

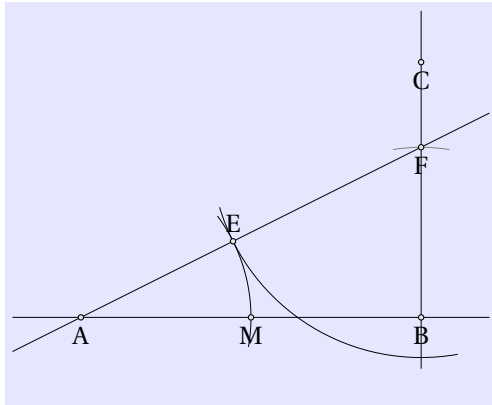


```
10 \begin{tikzpicture}
11   \tkzSetUpCompass[color=purple,line width=1pt]
12   \tkzDefPoints{0/1/A,5/2/B,3/4/C}
13   \tkzDefEquiPoints[from=C,dist=1,show,
14     /tkzcompass/delta=20] (A,B)
15   \tkzGetPoints{E}{H}
16   \tkzDrawLines[color=blue] (C,E C,H A,B)
17   \tkzDrawPoints[color=blue] (A,B,C)
18   \tkzDrawPoints[color=red] (E,H)
19   \tkzLabelPoints(E,H)
20   \tkzLabelPoints[color=blue] (A,B,C)
21 \end{tikzpicture}
```

9 Size option for code fonts

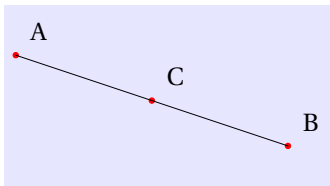
The different size options for fonts are :

1. very small



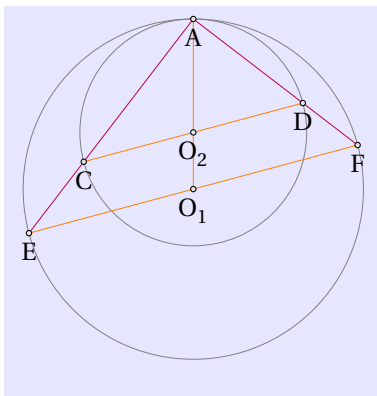
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{O/Q/A,6/Q/B}
  \tkzDefMidPoint(A,B)
  \tkzGetPoint{I}
  \tkzDefPointWith[orthogonal,K=-.75](B,A)
  \tkzGetPoint{C}
  \tkzInterLC(B,C)(B,I)
  \tkzGetPoints{F}{D}
  \tkzDuplicateSegment(B,F)(A,F)
  \tkzGetPoint{E}
  \tkzDrawArc[delta=10](F,E)(B)
  \tkzInterLC(A,B)(A,E)
  \tkzGetPoints{N}{M}
  \tkzDrawArc[delta=10](A,M)(E)
  \tkzDrawLines(A,B,B,C,A,F)
  \tkzCompass(B,F)
  \tkzDrawPoints(A,B,C,F,M,E)
  \tkzLabelPoints(A,B,C,F,M)
  \tkzLabelPoints[above](E)
\end{tikzpicture}
```

2. small



```
\begin{tikzpicture}[scale=1.2]
  \tkzDefPoints{1/3/A,4/2/B}
  \tkzDefPointWith[linear,K=0.5](A,B)
  \tkzGetPoint{C}
  \tkzDrawPoints[color=red](A,B,C)
  \tkzDrawSegment(A,B)
  \tkzLabelPoints[above right=3pt](A,B,C)
\end{tikzpicture}
```

3. normal



```
\begin{tikzpicture}[scale=.75]
\tikzset{new/.style={color=orange,
line width=.2pt}}
\tkzDefPoints{O/O_0_1,O/1/O_2,O/3/A}
\tkzDefPoint(15:3){F}
\tkzInterLC(F,O_1)(O_1,A)
\tkzGetSecondPoint{E}
\tkzDefLine[parallel=through O_2](E,F)
\tkzGetPoint{x}
\tkzInterLC(x,O_2)(O_2,A) \tkzGetPoints{D}{C}
\tkzDrawCircles(O_1,A O_2,A)
\tkzDrawSegments[new](O_1,A E,F C,D)
\tkzDrawSegments[purple](A,E A,F)
\tkzDrawPoints(A,O_1,O_2,E,F,C,D)
\tkzLabelPoints(A,O_1,O_2,E,F,C,D)
\end{tikzpicture}
```

10 Option code only

```

\begin{tkzexample}[small,code only]
\begin{minipage}{9cm}
\thispagestyle{empty}
\noindent
\begin{tikzpicture}
\draw (0,0) node[circle,
                shade,
                ball color=Peach,minimum size=2cm]{};
\draw[snake      = expanding waves,
      color      = Peach,
      line width = 3pt](1.2,0)--(4,0);
\end{tikzpicture}
\end{minipage}
\end{tkzexample}

```

The result

```

\begin{minipage}{9cm}
\begin{tikzpicture}
\draw (0,0) node[circle,
                shade,
                ball color=Peach,minimum size=2cm]{};
\draw[snake      = expanding waves,
      color      = Peach,
      line width = 3pt](1.2,0)--(4,0);
\end{tikzpicture}
\end{minipage}

```

11 Long code: option break code

Up to version 1.45c, the code placed beside (or under) the result was set in an unbreakable box: a code longer than the remaining space on the page went to the next page, and a code longer than a page overflowed at the bottom of the page and was lost.

Since version 1.51, the option `break code` allows the code to be split across pages. When the code must be split, the result is placed first, then the code under the result, in a frame that can be broken (with `overhang`, the code first then the result). The line numbers (option `num`), the background color (option `code`) and the frame (option `frame code`) continue on the following pages.

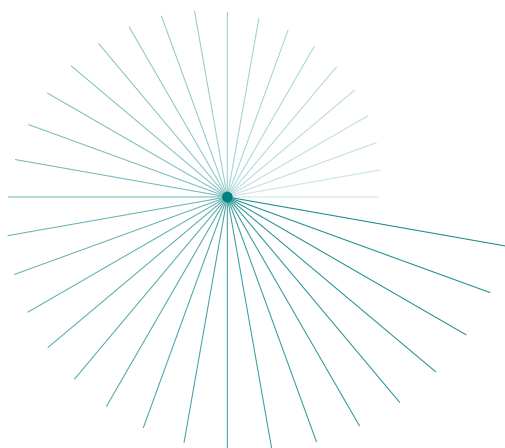
<code>break code=auto</code>	default: the code is split only if the example is higher than a page (<code>\textheight</code>);
<code>break code=fill</code>	the code is split if the example does not fit in the space left on the current page;
<code>break code</code>	same as <code>break code=fill</code> ;
<code>break code=always</code>	the result first, then the breakable code under it;
<code>break code=never</code>	the behaviour of the version 1.45c (unbreakable box).

With the default value `auto`, the layout of the examples that fit on a page does not change. To use `fill` for all the examples of a document:

```
\pgfkeys{/tkzexample/every tkzexample/.append style={break code}}
```

The next example is obtained with:

```
\begin{tkzexample}[break code,num,code=teal!10,graphic=white]
...
\end{tkzexample}
```



```
1 \begin{tikzpicture}
2   \draw[teal!20] (0,0) -- (0:2.00);
3   \draw[teal!22] (0,0) -- (10:2.05);
4   \draw[teal!24] (0,0) -- (20:2.10);
5   \draw[teal!26] (0,0) -- (30:2.15);
6   \draw[teal!28] (0,0) -- (40:2.20);
7   \draw[teal!30] (0,0) -- (50:2.25);
8   \draw[teal!32] (0,0) -- (60:2.30);
```

```

9 \draw[teal!34] (0,0) -- (70:2.35);
10 \draw[teal!36] (0,0) -- (80:2.40);
11 \draw[teal!38] (0,0) -- (90:2.45);
12 \draw[teal!40] (0,0) -- (100:2.50);
13 \draw[teal!42] (0,0) -- (110:2.55);
14 \draw[teal!44] (0,0) -- (120:2.60);
15 \draw[teal!46] (0,0) -- (130:2.65);
16 \draw[teal!48] (0,0) -- (140:2.70);
17 \draw[teal!50] (0,0) -- (150:2.75);
18 \draw[teal!52] (0,0) -- (160:2.80);
19 \draw[teal!54] (0,0) -- (170:2.85);
20 \draw[teal!56] (0,0) -- (180:2.90);
21 \draw[teal!58] (0,0) -- (190:2.95);
22 \draw[teal!60] (0,0) -- (200:3.00);
23 \draw[teal!62] (0,0) -- (210:3.05);
24 \draw[teal!64] (0,0) -- (220:3.10);
25 \draw[teal!66] (0,0) -- (230:3.15);
26 \draw[teal!68] (0,0) -- (240:3.20);
27 \draw[teal!70] (0,0) -- (250:3.25);
28 \draw[teal!72] (0,0) -- (260:3.30);
29 \draw[teal!74] (0,0) -- (270:3.35);
30 \draw[teal!76] (0,0) -- (280:3.40);
31 \draw[teal!78] (0,0) -- (290:3.45);
32 \draw[teal!80] (0,0) -- (300:3.50);
33 \draw[teal!82] (0,0) -- (310:3.55);
34 \draw[teal!84] (0,0) -- (320:3.60);
35 \draw[teal!86] (0,0) -- (330:3.65);
36 \draw[teal!88] (0,0) -- (340:3.70);
37 \draw[teal!90] (0,0) -- (350:3.75);
38 \fill[teal] (0,0) circle (2pt);
39 \end{tikzpicture}

```

12 Options of the frame of code only and tkzltxexample

With the options `code only` or with the environment `tkzltxexample`, the code is placed in a frame (`mdframed`). The options of the frame are: `left margin`, `right margin`, `inner left margin`, `inner right margin`, `inner top margin`, `inner bottom margin` and `line frame width`. Since version 1.50, `inner right margin` (default 10pt) and `inner bottom margin` (default `.4\baselineskip`) are taken into account.

```
\begin{tkzexample}[code only,line frame width=1pt,inner left margin=6pt,
                    inner bottom margin=6pt]
```

```
\tkzDefPoints{0/0/A,4/0/B}
\tkzDrawSegment(A,B)
```

13 Option saved of the package

With `\usepackage[saved]{tkzexample}`, the code of each example is written in a file and then read. The name of the file is `tkzFile-<section>-<subsection>-<subsubsection>.tex`. Since version 1.50, when several examples are in the same subsubsection, the files are no longer overwritten: the following files are named `tkzFile-1-2-0-2.tex`, `tkzFile-1-2-0-3.tex`, etc. The prefix `tkzFile` can be changed with `\tkzSavedCode[prefix=...]`.

14 History

Version 1.51 --- 2026/10/05

- New option `break code` (auto, fill, always, never): a long code can be split across pages.
- The option `above skip` works (it had no effect).
- The options may contain `\par` (for example `above skip=\par\vspace{1cm}`).

Version 1.50 --- 2026/10/05

- The packages `xcolor` and `pgfkeys` are loaded: `tkzexample` can be used without `tikz`.
- The brackets `[]` are optional.
- `overhang=false` works.
- The options `inner right margin` and `inner bottom margin` are used.
- `tkzltxexample`: the option `hsep` works.
- Options `latex` and `vbox`: an end of line in the code is a space and an empty line starts a new paragraph (the words of two lines were glued together).
- Option `saved`: with `code only` the code is printed (and not the name of the file); no more overflow of 6pt when the code is under the result; the saved files are no longer overwritten.
- `\tkzSavedCode` works without brackets.