The Ti*k*Z-Extensions Package

Manual for version 0.6 (8)

`https://github.com/Qrrbrbirlbel/tikz-extensions`

Qrrbrbirlbel

September 10, 2024

Contents

Part I
Introduction

1 Usage

This package is called `tikz-ext`, however, one can't load it via `\usepackage`.[1] Instead, this package consists mostly of PGF and Ti*k*Z libraries which are loaded by either `\usepgflibrary` or `\usetikzlibrary`.

2 Why do we need it?

Since I have been answering questions on TeX.sx I've noticed that some questions come up again and again, every time with a slightly different approach on how to solve them.

I don't like reinventing the wheel which is why I've gathered the solutions of my answers in this package.

3 Having problems?

Note however, that most of these extensions haven't been stress-tested properly and might be considered experimental.

Don't hesitate to open an issue on GitHub. You probably found a bug.

[1] Except for `pgfcalendar-ext` and `pgffor-ext`.

4 Namespaces and Ti*k*Z-Extensions macros

Since some parts of this package have existed in some form since 2013, the choice for key names and in which PGFkeys namespace they reside is not always optimal. They often reside in the main `/tikz` or `/pgf` path. Similar applies to macro names.

For future versions, it is planned to move those in the `/tikz/ext` namespace. For keys in the `/pgf` namespace, this will probably not happen since it makes it not very intuitive to use them in Ti*k*Z.

Starting from version 0.6, Ti*k*Z-Extensions provides commands that return the current version for compatibility testing. The second simply increments with every release so that the first doesn't need to be parsed.

`\tikzextversion`

Returns `0.6`.

`\tikzextversionnumber`

Returns `8`.

Also starting from version 0.6, there's `\tikzextset` and `\pgfextset`.

`\tikzextset{`⟨*options*⟩`}`

This command will process the ⟨*options*⟩ using the `\pgfkeys` command with the default path set to `/tikz/ext`.

`\pgfextset{`⟨*options*⟩`}`

This command will process the ⟨*options*⟩ using the `\pgfkeys` command with the default path set to `/pgf/ext`.

5 Compatibility with older versions

As discussed in the previous section, keys and commands of extensions that existed before version 0.6 that do not appear in this manual are considered deprecated.

`/tikz/ext/compat=pre 0.6|0.6|warn|newest` (default `pre 0.6`)

This sets the global compatibility setting for every extension of this package (whether already loaded or not).

The choice `warn` gives out warning for deprecated keys or commands but still executes them if they were not not in use when an extension was loaded.

For version 0.6 this is actually the default settings so that active documents keep working – for now.

The following table shows the compatibility settings for each extension. A ✓ denotes an available setting where ✓✓ denotes the default compatibility setting. A – denotes that it is not different than the `newest` setting.

Extension	warn	pre 0.6	0.6
`pgfcalendar-ext` `ext.calendar-plus`	✓✓	✓	–
`ext.arrows`	✓✓	✓	–
`ext.layers`	✓✓	✓	–
`ext.node-families`	✓✓	✓	–
`ext.nodes`	✓✓	✓	–
`ext.paths.arcto`	✓✓	✓	–
`ext.paths.ortho`	✓✓	✓	–
`ext.paths.timer`	✓✓	✓	–
`ext.pgffor`	✓✓	✓	–
`ext.positioning-plus`	✓✓	✓	–
`ext.scalepicture`	✓✓	✓	–
`ext.shapes`	✓✓	✓	–
`ext.transformations.mirror`	✓✓	✓	–
`ext.topaths.arcthrough`	✓✓	✓	–

For each available extension the compatibility setting can be adjusted as well after the extension is loaded.

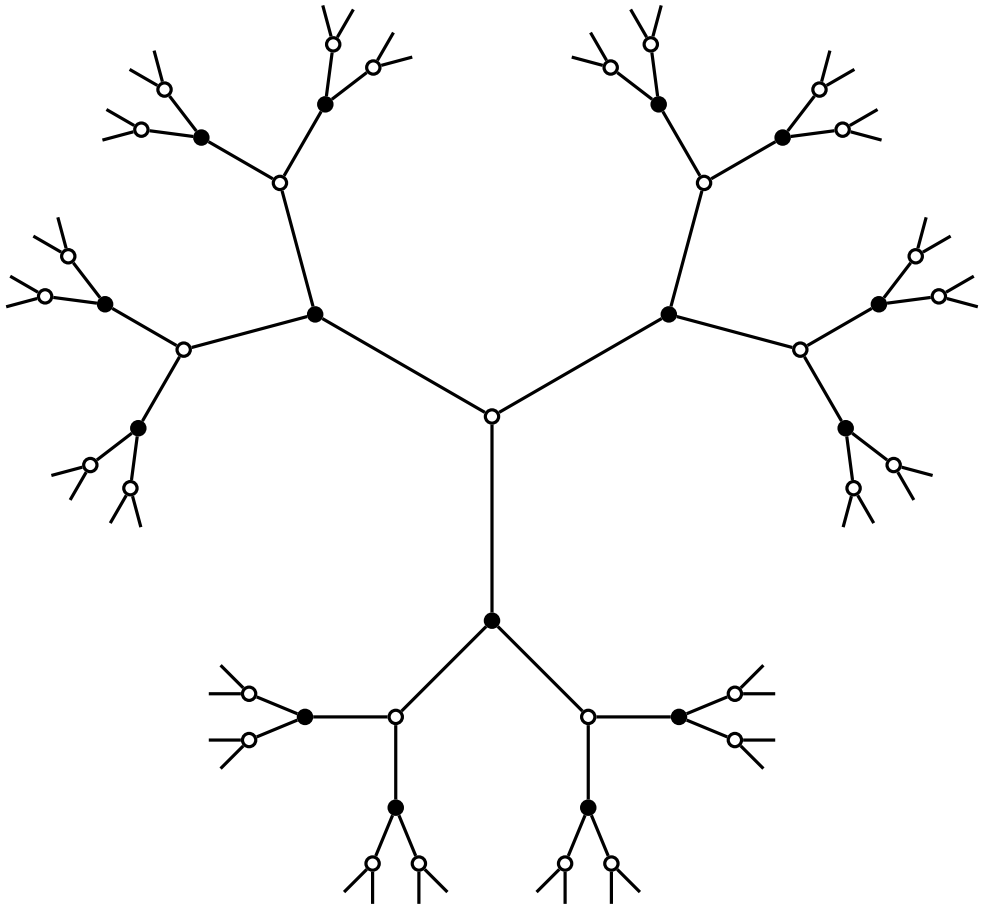
`/tikz/ext/compat/pgfcalendar-ext=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/arrows=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/layers=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/nodes=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/node-families=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/paths.arcto=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/paths.ortho=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/paths.timer=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/pgffor=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/positioning-plus=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/scalepicture=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/shapes=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/transformations.mirror=`⟨*version*⟩ (default `pre 0.6`)
`/tikz/ext/compat/topaths.arcthrough=`⟨*version*⟩ (default `pre 0.6`)

For ⟨*version*⟩ the same choices are valid as for the main `compat` key. It should be noted that at this point, a compatibility setting can't really be reversed since they only forward arguments from an old key or command to the new version.

The old names are given as a subtitle to the new one in the sections that introduce them.

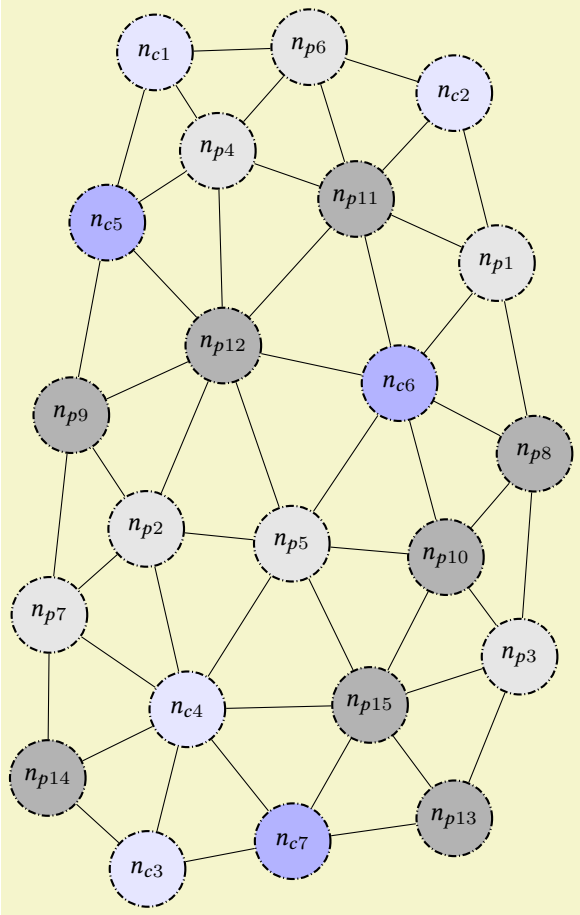
Part II
Ti*k*Z Libraries

These libraries only work with Ti*k*Z.



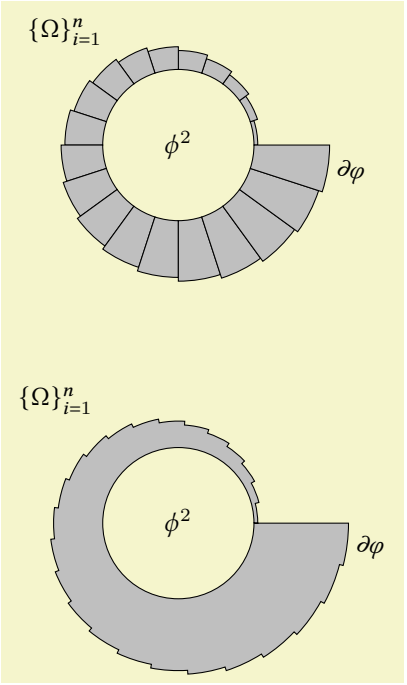
Part III
PGF Libraries

These libraries (should) work with both PGF and Ti*k*Z.




```
\usetikzlibrary {graphs,graphdrawing,ext.misc} \usegdlibrary {force}
\tikzset{
  mynode/.style={
    circle, minimum size=10mm, draw, densely dashdotted, thick,
    decide color/.expand once=#1},
  decide color/.style 2 args={
    /utils/TeX/if=c#1
      {/utils/TeX/ifnum={#2<5}{bluelight}{bluedark}}
      {/utils/TeX/ifnum={#2<8}{light}{dark}}},
  light/.style={fill=gray!20},  bluelight/.style={fill=blue!10},
  dark/.style ={fill=gray!60},  bluedark/.style ={fill=blue!30}}
\tikz\graph[
  spring electrical layout, vertical=c2 to p13,
  node distance=1.5cm, typeset=$n_{\tikzgraphnodetext}$,
  nodes={mynode=\tikzgraphnodetext}] {
  % outer ring
  c2 -- {p1, p11, p6};
    p1 -- {p8, c6, p11};
      p8 -- {p3, p10, c6};
       p3 -- {p13, p15, p10};
         p13 -- {p15, c7};
           c7  -- {c3, c4, p15};
           c3  -- {p14, c4};
           p14 -- {p7, c4};
         p7 -- {p9, p2, c4};
       p9 -- {c5, p12, p2};
     c5 -- {c1, p4, p12};
   c1 -- {p6, p4};
  p6 -- {p11, p4};
  % inner ring
  p11 -- {c6, p12, p4};
  p5 -- {c6 -- {p10, p12}, p10 -- p15, p15 -- c4, c4 -- p2, p2 -- p12, p12 -- p4};
};
```

Part IV
Utilities




```
\usetikzlibrary {ext.misc}
\begin{tikzpicture}[
  declare function={bigR(\n)=smallR+.05*\n;},
  declare constant={smallR=1; segments=20;},
  full arc=segments]
\foreach \iN[evaluate={\endRadius=bigR(\iN+1);}, ext/use int=0 to segments-1]
  \filldraw[fill=gray!50] (\iN R:\endRadius)
    arc [radius=\endRadius, start angle=\iN R, delta angle=+1R] -- (\iN R+1R:smallR)
    arc [radius=smallR,       end angle=\iN R, delta angle=-1R] -- cycle;

\node                                              {$\phi^2$};
\node at (north west:{sqrt 2 * bigR(segments/2)})  {$\{\Omega\}_{i=1}^n$};
\node[rotate=-.5R, right] at (-.5R: bigR segments) {$\partial \varphi$};

\tikzset{yshift=-5cm, declare constant={segments=25;}, full arc=segments}
\filldraw[fill=gray!50] (right:smallR)
  \foreach \iN[evaluate={\endRadius=bigR(\iN+1);}, ext/use int=0 to segments-1] {
    -- (\iN R:\endRadius) arc[radius=\endRadius, start angle=\iN R, delta angle=1R]}
    -- (right:smallR)     arc[radius=smallR,     start angle=0,     delta angle=-360];

\node                                              {$\phi^2$};
\node at (north west:{sqrt 2 * bigR(segments/2)})  {$\{\Omega\}_{i=1}^n$};
\node[rotate=-.5R, right] at (-.5R: bigR segments) {$\partial \varphi$};
\end{tikzpicture}
```

29 Repeating Things and Other Things

```
\usepackage{pgffor-ext} % LaTeX
\input pgffor-ext.tex   % plain TeX
```

This package adds small niceties to the `pgffor` package. Most of these additions are also available with the `ext.misc` library.

Warning: Consider this package experimental. At the very least, it will break the `...` notation and possibly gobbles spaces after the body.

Q & A: [2, 8, 56] & [38, 44, 40]

Instead of `\foreach \var in {start, start + delta, ..., end}` one can use `\foreach \var[use int=start to end step delta]`.

`/pgf/foreach/ext/use int=`⟨*start*⟩`to`⟨*end*⟩`step`⟨*delta*⟩ (no default)

pre 0.6 `/pgf/foreach/use int`

The values ⟨*start*⟩, ⟨*end*⟩ and ⟨*delta*⟩ are evaluates by PGFmath at initialization. The part `step` ⟨*delta*⟩ is optional (⟨*delta*⟩ = 1).

`/pgf/foreach/ext/use float=`⟨*start*⟩`to`⟨*end*⟩`step`⟨*delta*⟩ (no default)

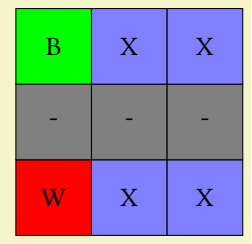
pre 0.6 `/pgf/foreach/use float`

Same as above, however the results are not truncated.

`/pgf/foreach/ext/no separator` (no value)

pre 0.6 `/pgf/foreach/no separator`

This key disables any separator between elements of the list. Every token is its own element. This also means that Unicode characters need to be grouped between `{` and `}` if LuaTeX isn't used. Spaces will be ignored.




```
\usetikzlibrary {ext.misc}
\newcommand*{\board}[3][]{%
  \begin{tikzpicture}[#1]
    \foreach[
      count=\i from 0,
      ext/no separator,
      evaluate=\i as \colX using {mod(\i,#2)},
      evaluate=\i as \rowY using {int(\i/#2)}
    ] \elem in {#3} {
        \draw[black, board/\elem/.try, ext/rectangle timer/.try=line]
          (\colX,\rowY) rectangle node {\elem} ++(1, 1);}
  \end{tikzpicture}}
\board[
  board/W/.style={fill=red},
  board/X/.style={fill=blue!50},
  board/B/.style={fill=green},
  board/-/.style={fill=gray},
]{3}{WXX---BXX}
```

`/pgf/foreach/ext/normal list` (no value)

pre 0.6 `/pgf/foreach/normal list`

This key simply disables all other special parsers and returns to the original list parser.

The following keys only work with LaTeX and cannot be used when only the `ext.misc` library or the plainTeX `pgffor-ext.tex` are loaded. For this, you will need to use `\usepackage{pgffor-ext}`.

`/pgf/foreach/ext/xparser`={⟨*argument specification*⟩}{⟨*foreach value*⟩} (no default)

pre 0.6 `/pgf/foreach/xparser`

This key can be used to specify a `xparse` specification for each element in the list.

For this to work somewhat seamless, the following needs to observed:

- Every {⟨*argument specification*⟩} get appended `u,`. This means there's always one additional mandatory argument at the end of every element.
- The {⟨*foreach value*⟩} needs to correspond to the `/pgf/foreach/var` value.

`/pgf/foreach/ext/xparser Om`=default (default {})

pre 0.6 `/pgf/foreach/xparser Om`

Sets up a list whose elements may contain an optional argument inside `[]` which correspond to two `\foreach` variables, say `\Options/\Text`. The default value is the default value if the optional argument is missing.

Key handler ⟨*key*⟩`/.ext list xparse`={⟨*argument specification*⟩}{⟨*comma-separated list of values*⟩}

pre 0.6 `.list xparse`

This handler causes the key to be used repeatedly, namely once for every element of the list of values. The ⟨*comma-separated list of values*⟩ is processed using `\foreach` and the given `xparse` ⟨*argument specification*⟩ with the aforementioned `xparser` key.

30 And a little bit more

TikZ Library `ext.misc`

```
\usetikzlibrary{ext.misc} % LaTeX and plain TeX
\usetikzlibrary[ext.misc] % ConTeXt
```

This library adds miscellaneous utilities to PGFmath, PGF or TikZ.

Q & A: [24] & [27]

30.1 PGFmath

30.1.1 Postfix operator R

Similar to `\segments[<num>]` in PSTricks, the postfix operator `R` allows the user to use an arbitrary number of segments of a circle to be used instead of an angle.

`/pgf/full arc=`⟨*num*⟩ (default `{}`)

The number ⟨*num*⟩ of segments will be set up. Using `full arc` with an empty value disables the segmentation and `1R` equals 1°.

The given value ⟨*num*⟩ is evaluated when the key is used and doesn't change when ⟨*num*⟩ contains variables that change.

The `R` operator can then be used.

x`R` (postfix operator; uses the `fullarc` function)

Multiplies *x* with $\frac{360}{\langle num \rangle}$.

30.1.2 Functions

`strrepeat("`*Text*`", `*x*`)`
`\pgfmathstrrepeat{"`*Text*`"}{`*x*`}`

Returns a string with *Text* repeated *x* times.

foofoofoofoofoo

```
\pgfmathparse{strrepeat("foo", 5)}
\pgfmathresult
```

`isInString("`*String*`", "`*Text*`")`
`\pgfmathisInString{"`*String*`"}{"`*Text*`"}`

Returns `1` (true) if *Text* contains *String*, otherwise `0` (false).

0 and 1

```
\pgfmathparse{isInString("foo", "bar")}
\pgfmathresult \ and\
\pgfmathparse{isInString("foo", "foobar")}
\pgfmathresult
```

`strcat("`*Text A*`", "`*Text B*`", …)`
`\pgfmathstrcat{"`*Text A*`"}{"`*Text B*`"}{…}`

Returns the concatenation of all given parameters.

blue!21!green

```
\pgfmathparse{strcat("blue!", int(7*3), "!green")}
\pgfmathresult
```

`isEmpty("`*Text*`")`
`\pgfmathisEmpty{"`*Text*`"}`

Returns `1` (true) if *Text* is empty, otherwise `0` (false).

0 and 1 and 1

```
\pgfmathparse{isEmpty("foo")} \pgfmathresult\ and\
\pgfmathparse{isEmpty("")}    \pgfmathresult\ and\
\def\emptyText{}
\pgfmathparse{isEmpty("\emptyText")} \pgfmathresult
```

`atanXY(`*x*`,`*y*`)`

`\pgfmathatanXY{`*x*`}{`*y*`}`

Arctangent of $y \div x$ in degrees. This also takes into account the quadrant. This is just a argument-swapped version of `atan2` which makes it easier to use the `\p` commands of the `calc` library.

53.13011 `\pgfmathparse{atanXY(3,4)} \pgfmathresult`

`atanYX(`*y*`,`*x*`)`
`\pgfmathatanYX{`*y*`}{`*x*`}`

Arctangent of $y \div x$ in degrees. This also takes into account the quadrant.

53.13011 `\pgfmathparse{atanYX(4,3)} \pgfmathresult`

30.1.3 Functions: using coordinates

The following functions can only be used with PGF and/or Ti*k*Z. Since the arguments are usually plain text (and not numbers) one has to wrap them in `"`.

`anglebetween("`*p1*`", "`*p2*`")`
`\pgfmathanglebetween{"`*p1*`"}{"`*p2*`"}`

Return the angle between the centers of the nodes *p1* and *p2*.

`qanglebetween("`*p*`")`
`\pgfmathqanglebetween{"`*p*`"}`

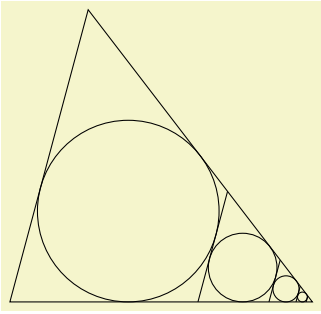
Return the angle between the origin and the center of the node *p*.

`distancebetween("`*p1*`", "`*p2*`")`
`\pgfmathdistancebetween{"`*p1*`"}{"`*p2*`"}`

Return the distance (in pt) between the centers of the nodes *p1* and *p2*.

`qdistancebetween("`*p*`")`
`\pgfmathqdistancebetween{"`*p*`"}`

Return the distance (in pt) between the origin and the center of the node *p*.



```
\usetikzlibrary {calc,ext.misc,through}
\begin{tikzpicture}
\path (0,0) coordinate (A) + (0:4) coordinate (B) +(75:4) coordinate (C);
\draw (A) -- (B) -- (C) -- cycle;
\foreach \cnt in {1,...,4}{
  \pgfmathsetmacro\triA{distancebetween("B","C")}
  \pgfmathsetmacro\triB{distancebetween("C","A")}
  \pgfmathsetmacro\triC{distancebetween("A","B")}
  \path (barycentric cs:A=\triA,B=\triB,C=\triC) coordinate (M)
       node [draw, circle through=($(A)!(M)!(C)$)] (M) {};
  \draw ($(C)-(A)$) coordinate (vecB)
      (M.75-90) coordinate (@)
      (intersection of @--[shift=(vecB)]@ and B--C) coordinate (C) --
      (intersection of @--[shift=(vecB)]@ and B--A) coordinate (A);}
\end{tikzpicture}
```

30.2 PGFfor

This library loads also most of the functions of the `pgffor-ext` of section 29 on page 72.

30.3 PGFkeys

pgfkeys Library `ext.pgfkeys-plus`

```
\usepgfkeyslibrary{ext.pgfkeys-plus} % LaTeX and plain TeX
\usepgfkeyslibrary[ext.pgfkeys-plus] % ConTeXt
```

This extends PGFkeys and adds helpful `/utils` keys as well as handlers. This library gets loaded by the `ext.misc` library.

30.3.1 Conditionals

`/utils/if`={⟨*cond*⟩}{⟨*true*⟩}{⟨*false*⟩} (no default)

This key checks the conditional ⟨*cond*⟩ and applies the styles ⟨*true*⟩ if ⟨*cond*⟩ is true, otherwise ⟨*false*⟩. ⟨*cond*⟩ can be anything that PGFmath understands.

As a side effect on how PGFkeys parses argument, the ⟨*false*⟩ argument is actually optional.

The following keys use TeX' macros `\if`, `\ifx`, `\ifnum` and `\ifdim` for faster executions.

`/utils/TeX/if`=⟨*token A*⟩⟨*token B*⟩{⟨*true*⟩}{⟨*false*⟩} (no default)

This key checks via `\if` if ⟨*token A*⟩ matches ⟨*token B*⟩ and applies the styles ⟨*true*⟩ if it does, otherwise ⟨*false*⟩.

As a side effect on how PGFkeys parses argument, the ⟨*false*⟩ argument is actually optional.

`/utils/TeX/ifx`=⟨*token A*⟩⟨*token B*⟩{⟨*true*⟩}{⟨*false*⟩} (no default)

As above but via `\ifx`.

`/utils/TeX/ifnum`={⟨*num cond*⟩}{⟨*true*⟩}{⟨*false*⟩} (no default)

This key checks `\ifnum`⟨*num cond*⟩ and applies the styles ⟨*true*⟩ if true, otherwise ⟨*false*⟩. A delimiting `\relax` will be inserted after ⟨*num cond*⟩.

As a side effect on how PGFkeys parses arguments, the ⟨*false*⟩ argument is actually optional.

`/utils/TeX/ifdim`=⟨*dim cond*⟩⟨*true*⟩⟨*false*⟩ (no default)

As above but with `\ifdim`.

`/utils/TeX/ifempty`=⟨*Text*⟩⟨*true*⟩⟨*false*⟩ (no default)

This checks whether ⟨*Text*⟩ is empty and applies styles ⟨*true*⟩ if true, otherwise ⟨*false*⟩.

`/utils/TeX/ifxempty`=⟨*Text*⟩⟨*true*⟩⟨*false*⟩ (no default)

This checks whether fully expanded ⟨*Text*⟩ is empty and applies styles ⟨*true*⟩ if true, otherwise ⟨*false*⟩.

30.3.2 Handlers

While already a lot of values given to keys are evaluated by PGFmath at some point, not all of them are.

Key handler ⟨*key*⟩`/.pgfmath`=⟨*eval*⟩

This handler evaluates ⟨*eval*⟩ before it is handed to the key.

This handler works almost the same as the `.evaluated` handler but it does its evaluation in a group so that the result will not overwrite any other results.

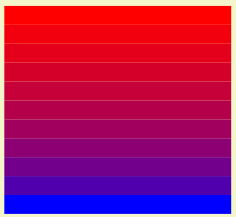
Key handler ⟨*key*⟩`/.pgfmath int`=⟨*eval*⟩

As above but truncates the result.

Key handler ⟨*key*⟩`/.pgfmath wrap`={⟨*wrapper*⟩}{⟨*eval*⟩}

This feeds the result of ⟨*eval*⟩ as `#1` to ⟨*wrapper*⟩.

In the example below, one could have used the `/pgf/foreach/evaluate` key from the `\foreach` loop.



```
\usepgfkeyslibrary {ext.pgfkeys-plus}
\tikz\foreach \i in {0,10,...,100}
\draw[
  line width=+.25cm,
  % needs ## because its inside the \foreach body
  color/.pgfmath wrap={red!##1!blue}{sqrt(\i)*10}
]
  (0,\i/40) -- +(right:3);
```

Key handler ⟨*key*⟩`/.pgfmath if=`{⟨*cond*⟩}{⟨*true*⟩}{⟨*false*⟩}

Evaluates ⟨*cond*⟩ with PGFMath and returns ⟨*true*⟩ or ⟨*false*⟩ to the used key respectively.

Key handler ⟨*key*⟩`/.if=`⟨*token A*⟩⟨*token B*⟩{⟨*true*⟩}{⟨*false*⟩}

Checks via `\if` if ⟨*token A*⟩ matches ⟨*token B*⟩ and applies the value ⟨*true*⟩ if it does, otherwise ⟨*false*⟩.

Key handler ⟨*key*⟩`/.ifx=`⟨*token A*⟩⟨*token B*⟩{⟨*true*⟩}{⟨*false*⟩}

As above but via `\ifx`.

Key handler ⟨*key*⟩`/.ifnum=`{⟨*ifnum cond*⟩}{⟨*true*⟩}{⟨*false*⟩}

Checks via `\ifnum` if ⟨*ifnum cond*⟩ and applies the value ⟨*true*⟩ if it does, otherwise ⟨*false*⟩.

Key handler ⟨*key*⟩`/.ifdim=`{⟨*ifdim cond*⟩}{⟨*true*⟩}{⟨*false*⟩}

As above but via `\ifdim`.

Key handler ⟨*key*⟩`/.ifxempty=`{⟨*Text*⟩}{⟨*true*⟩}{⟨*false*⟩}

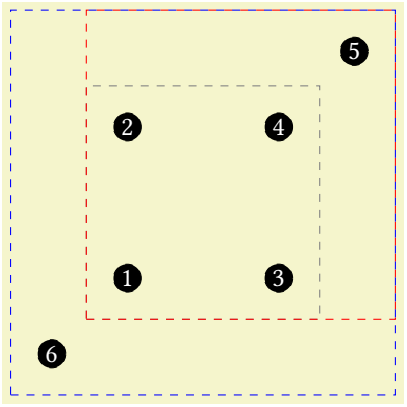
Checks whether a fully expanded ⟨*Text*⟩ is empty and applies the value ⟨*true*⟩ if it does, otherwise ⟨*false*⟩.

Key handler ⟨*key*⟩`/.ifempty=`{⟨*Text*⟩}{⟨*true*⟩}{⟨*false*⟩}

Checks whether ⟨*Text*⟩ is empty and applies the value ⟨*true*⟩ if it does, otherwise ⟨*false*⟩.

Key handler ⟨*key*⟩`/.List=`{⟨⟨*e1*⟩, ⟨*e2*⟩, ..., ⟨*en*⟩⟩}

This handler evaluates the given list with `\foreach` and concatenates the element and the result is then given to the used key.




```
\usetikzlibrary {fit,ext.misc}
\begin{tikzpicture}[nodes={draw, dashed, inner sep=+10pt}]
  \foreach \point [count=\cnt] in {(0,0), (0,2), (2,0), (2,2), (3,3), (-1,-1)}
    \node[circle, fill, inner sep=1pt, text=white] (point-\cnt) at \point {\cnt};
  \node[gray, fit/.List={(point-1),(point-...),(point-4)}] {};
  \node[red,  fit/.List={(point-1),(point-...),(point-5)}] {};
  \node[blue, fit/.List={(point-1),(point-...),(point-6)}] {};
\end{tikzpicture}
```

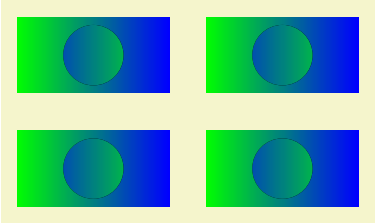
30.4 Ti*k*Z

`/tikz/reverse clip=`⟨*direction*⟩ (default `counter clockwise`)

This key installs a very big rectangle which is either constructed `counter clockwise` (like the `circle` path operation) or `clockwise`.

`/tikz/clip rule=`⟨*direction*⟩ (default `even odd`)

This key switches directly[8] to the specified rule which is either `even odd` or `nonzero`. This corresponds to the `/tikz/even odd rule` and `/tikz/nonzero rule` keys.



```
\usetikzlibrary {ext.misc}
\newcommand*\myDiagram[1]{
  \fill[left color=blue, right color=green] (0, 0) rectangle (2, 1);
  \clip (1, .5) #1 [reverse clip];
  \fill[left color=green, right color=blue] (0, 0) rectangle (2, 1);
}
\begin{tikzpicture}[radius=.4, row sep=5mm, column sep=5mm]
\matrix[
  row 2/.append style={clip rule=even odd},
  column 1/.append style={reverse clip/.default=clockwise}
]{
  \myDiagram{circle[]} &
  \myDiagram{+(0:.4) arc[start angle=0, delta angle=-360] -- cycle}
\\
  \myDiagram{circle[]} &
  \myDiagram{+(0:.4) arc[start angle=0, delta angle=-360] -- cycle}
\\};
\end{tikzpicture}
```

[8] Meaning, it directly executes `\pgfseteorule` /`\pgfsetnonzerorule` and doesn't accumulates where Ti*k*Z throws an error.

Part V
Changelog, Index & References

Changelog

Version 0.6 (2024-09-10)

- Added `\tikzextset`, `\tikzextversion` and `\tikzextversionnumber`
- Added six new `auto` placement mechanisms: `ext/above`, `ext/below`, `ext/west`, `ext/east`, `ext/north` and `ext/south`.
- Added `ext/auto offset` for `auto` placement.
- Added `ext/precise auto angle`.
- Added Ti*k*Z library `ext.arrows-plus`.
- Added Ti*k*Z library `ext.topaths.autobend`.
- Made `ext.node-families` and `ext.scalepicture` memoizable.

Version 0.5.1 (2023-04-02)

- Added PGF library `ext.arrows`.
- Bugfix to `ext.pgfkeys-plus`. [21]

Version 0.5 (2023-03-17)

- Added package `pgffor-ext`.
- Added Ti*k*Z library `ext.nodes`.
- Added Ti*k*Z library `ext.layers`.
- Bugfixes to `ext.calendar-plus`.
- Allow the original `rectangle` timer with `ext.paths.timer`.

Version 0.4.2 (2022-10-30)

- Added Ti*k*Z library `ext.scalepicture`.
- Bugfixes to `shapes.uncenteredrectangle`, `paths.ortho`, `positioning-plus` and `pgfcalender-ext`.

Version 0.4.1 (2022-10-23)

- Cleaned up directory structure of documentary.
- Added PGFkeys library `ext.pgfkeys-plus`.
- Added shape `uncentered rectangle` (PGF library `ext.shapes.uncenteredrectangle`).
- Fixed `ext.paths.arcto` – again [20].

Version 0.4 (2022-10-10)

- CTAN version of 0.3.1

Version 0.3.1 (2022-10-09)

- Fixed `ext.paths.ortho` keys `only vertical first` and `only horizontal first`.
- Moved all (except the `to paths`) to namespace `/tikz/ortho`. `/tikz/hvvh` and `/tikz/udlr` are considered deprecated.
- Fixed `\pgfcalendarjulianyeartoweek`.
- Added more calendar tests.
- Added directory structure.

Version 0.3 (2022-09-24)

- Added shape `circle arrow` (PGF library `ext.shapes.circlearrow`).

- Added shape `circle cross split` (PGF library `ext.shapes.circlecrosssplit`).
- Added shape `heatmark` (PGF library `ext.shapes.heatmark`).
- Added shape `rectangle with rounded corners` (PGF library `ext.shapes.rectangleroundedcorners`).
- Added shape `superellipse` (PGF library `ext.shapes.superellipse`).
- Added Ti*k*Z library `ext.node-families.shapes.geometric`.
- Fixed `ext.node-families`' key `size`.
- Renamed internal macros to use custom namespace starting with `\tikzext@`.
- Added some references.

Version 0.2 (2022-08-21)

- Added Ti*k*Z library `ext.positioning-plus`.
- Added Ti*k*Z library `ext.node-families`.

Version 0.1 (2022-08-16)

- Added Ti*k*Z library `ext.calendar-plus`.
- Added Ti*k*Z library `ext.misc`.
- Added Ti*k*Z library `ext.paths.arcto`.
- Added Ti*k*Z library `ext.paths.ortho`.
- Added Ti*k*Z library `ext.paths.timer`.
- Added Ti*k*Z library `ext.patterns.images`.
- Added Ti*k*Z library `ext.topaths.arcthrough`.
- Added Ti*k*Z library `ext.transformations.mirror`.
- Added PGF library `ext.transformations.mirror`.

Index

This index contains automatically generated entries as well as references to original functionalities of PGF/Ti*k*Z and references to functionalities outside of PGF/Ti*k*Z.

References

[1] *'sloped' should consider the current transformation · Issue #1058 · pgf-tikz/pgf*. URL: https://github.com/pgf-tikz/pgf/issues/1058 (visited on 10/21/2023).

[2] Foo Bar. *How to use declared TikZ functions in \foreach condition?* TeX - LaTeX Stack Exchange. Apr. 2013. URL: https://tex.stackexchange.com/q/110962 (visited on 09/24/2022) (cit. on p. 72).

[3] boje. *Heatmap over country like Google Map*. May 2013. URL: https://tex.stackexchange.com/q/112929 (visited on 09/24/2022).

[4] Christian. *TikZ arrow tip is displaced*. TeX - LaTeX Stack Exchange. Apr. 2013. URL: https://tex.stackexchange.com/q/111051 (visited on 04/02/2023).

[5] cis. *TikZ / calendar: Set the height of a monthly calendar*. Dec. 2018. URL: https://tex.stackexchange.com/q/464589 (visited on 09/24/2022).

[6] cis. *TikZ: How to place a coordinate at parabola-path-position?* May 2020. URL: https://tex.stackexchange.com/q/543251 (visited on 09/24/2022).

[7] CrazyArm. *Is It Possible to Combine TikZ Distance and Line-To Operations?* Apr. 2013. URL: https://tex.stackexchange.com/q/106558 (visited on 09/24/2022).

[8] daan. *String conditional tikz*. TeX - LaTeX Stack Exchange. Nov. 2022. URL: https://tex.stackexchange.com/q/666263 (visited on 12/03/2022) (cit. on p. 72).

[9] Alejandro DC. *Better fitting line to node in TiKZ*. TeX - LaTeX Stack Exchange. Apr. 2015. URL: https://tex.stackexchange.com/q/241074 (visited on 04/01/2023).

[10] Dimitris. *Draw two concentric circles and a shaded area with associated text*. TeX - LaTeX Stack Exchange. Dec. 2022. URL: https://tex.stackexchange.com/q/667338 (visited on 12/12/2022).

[11] Fence. *Add week day to calendar*. Nov. 2019. URL: https://tex.stackexchange.com/q/517338 (visited on 09/24/2022).

[12] healyp. *TikZ calendar and conditional tests*. Oct. 2013. URL: https://tex.stackexchange.com/q/140948 (visited on 09/24/2022).

[13] Jan Hlavacek. *Modifying * and o style tikz arrows so that they are centered at the end of line*. TeX - LaTeX Stack Exchange. Feb. 2011. URL: https://tex.stackexchange.com/q/11871 (visited on 04/02/2023).

[14] Holene. *Dependent node size in TikZ*. Apr. 2017. URL: https://tex.stackexchange.com/q/107227 (visited on 09/24/2022).

[15] Edgar A. Bering IV. *Set the color of a tikz-cd Glyph arrow tip with xelatex*. TeX - LaTeX Stack Exchange. Oct. 2020. URL: https://tex.stackexchange.com/q/565010 (visited on 04/01/2023).

[16] jd6. *Full weeks in Tikz Calendar*. TeX - LaTeX Stack Exchange. Dec. 2020. URL: https://tex.stackexchange.com/q/576673 (visited on 10/09/2022).

[17] knut. *TikZ: Define pattern with reference to external picture*. Mar. 2013. URL: https://tex.stackexchange.com/q/103980 (visited on 09/24/2022).

[18] Ben Liblit. *path with both mark connection node and arrow tip*. TeX - LaTeX Stack Exchange. Feb. 2013. URL: https://tex.stackexchange.com/q/99945 (visited on 12/12/2022).

[19] Marco. *TikZ - Four Colored Circle Split*. Apr. 2017. URL: https://tex.stackexchange.com/q/121686 (visited on 09/24/2022).

[20] marmotghost. *clockwise/counter clockwise does not seem to work reliably*. Oct. 2022. URL: https://github.com/Qrrbrbirlbel/tikz-extensions/issues/2 (visited on 10/23/2022) (cit. on p. 79).

[21] marmotghost. *Latest version of ext.misc on CTAN appears to have a typo*. Mar. 2023. URL: https://github.com/Qrrbrbirlbel/tikz-extensions/issues/6 (visited on 04/01/2023) (cit. on p. 79).

[22] Alan Munn. *Determine TikZ bend direction automatically*. TeX - LaTeX Stack Exchange. Oct. 2023. URL: https://tex.stackexchange.com/q/699883 (visited on 10/31/2023).

[23] nkk. *How to prevent tikz custom node fill from covering the text when using node-families library*. June 2019. URL: https://tex.stackexchange.com/q/494862 (visited on 09/24/2022).

[24] Anthony Peter. *A rather difficult ring like picture to be drawn*. Apr. 2017. URL: https://tex.stackexchange.com/q/144293 (visited on 09/24/2022) (cit. on p. 74).

[25] projetmbc. *forest - automatic setting of the alignment of some labels*. TeX - LaTeX Stack Exchange. Oct. 2022. URL: https://tex.stackexchange.com/q/661726 (visited on 10/23/2022).

[26] projetmbc. *TikZ - "Circled" arrow*. TeX - LaTeX Stack Exchange. Jan. 2013. URL: https://tex.stackexchange.com/q/95221 (visited on 09/24/2022).

[27] Qrrbrbirlbel. *Answer to "A rather difficult ring like picture to be drawn"*. Nov. 2013. URL: https://tex.stackexchange.com/a/144297 (visited on 09/24/2022) (cit. on p. 74).

[28] Qrrbrbirlbel. *Answer to "Add week day to calendar"*. July 2022. URL: https://tex.stackexchange.com/a/651888 (visited on 09/24/2022).

[29] Qrrbrbirlbel. *Answer to "An oval surrounded a *long text* inside in TikZ [equivalent cover background of METAFUN]"*. TeX - LaTeX Stack Exchange. Aug. 2022. URL: https://tex.stackexchange.com/a/654759 (visited on 09/24/2022).

[30] Qrrbrbirlbel. *Answer to "Better fitting line to node in TiKZ"*. TeX - LaTeX Stack Exchange. Apr. 2015. URL: https://tex.stackexchange.com/a/241303 (visited on 04/01/2023).

[31] Qrrbrbirlbel. *Answer to "Dependent node size in TikZ"*. June 2013. URL: https://tex.stackexchange.com/a/121054 (visited on 09/24/2022).

[32] Qrrbrbirlbel. *Answer to "Determine TikZ bend direction automatically"*. TeX - LaTeX Stack Exchange. Oct. 2023. URL: https://tex.stackexchange.com/a/699919 (visited on 10/31/2023).

[33] Qrrbrbirlbel. *Answer to "Draw two concentric circles and a shaded area with associated text"*. TeX - LaTeX Stack Exchange. Dec. 2022. URL: https://tex.stackexchange.com/a/667341 (visited on 12/12/2022).

[34] Qrrbrbirlbel. *Answer to "forest - automatic setting of the alignment of some labels"*. TeX - LaTeX Stack Exchange. Oct. 2022. URL: https://tex.stackexchange.com/a/661746 (visited on 10/23/2022).

[35] Qrrbrbirlbel. *Answer to "Full weeks in Tikz Calendar"*. TeX - LaTeX Stack Exchange. Oct. 2022. URL: https://tex.stackexchange.com/a/660335 (visited on 10/09/2022).

[36] Qrrbrbirlbel. *Answer to "Heatmap over country like Google Map"*. May 2013. URL: https://tex.stackexchange.com/a/113004 (visited on 09/24/2022).

[37] Qrrbrbirlbel. *Answer to "How to draw a mixing rule? #chemistry"*. TeX - LaTeX Stack Exchange. Sept. 2022. URL: https://tex.stackexchange.com/a/657449 (visited on 10/23/2022).

[38] Qrrbrbirlbel. *Answer to "How to use declared TikZ functions in \foreach condition?"* TeX - LaTeX Stack Exchange. Apr. 2013. URL: https://tex.stackexchange.com/a/110996 (visited on 09/24/2022) (cit. on p. 72).

[39] Qrrbrbirlbel. *Answer to "Is It Possible to Combine TikZ Distance and Line-To Operations?"* Apr. 2013. URL: https://tex.stackexchange.com/a/106571 (visited on 09/24/2022).

[40] Qrrbrbirlbel. *Answer to "Is there a package to implement this style of "Register diagrams with field descriptions""*. TeX - LaTeX Stack Exchange. Dec. 2022. URL: https://tex.stackexchange.com/a/667155 (visited on 12/03/2022) (cit. on p. 72).

[41] Qrrbrbirlbel. *Answer to "Modifying * and o style tikz arrows so that they are centered at the end of line"*. TeX - LaTeX Stack Exchange. Sept. 2022. URL: https://tex.stackexchange.com/a/656241 (visited on 04/02/2023).

[42] Qrrbrbirlbel. *Answer to "path with both mark connection node and arrow tip"*. TeX - LaTeX Stack Exchange. Dec. 2022. URL: https://tex.stackexchange.com/a/667487 (visited on 12/12/2022).

[43] Qrrbrbirlbel. *Answer to "Set the color of a tikz-cd Glyph arrow tip with xelatex"*. TeX - LaTeX Stack Exchange. Apr. 2023. URL: https://tex.stackexchange.com/a/681474 (visited on 04/01/2023).

[44] Qrrbrbirlbel. *Answer to "String conditional tikz"*. TeX - LaTeX Stack Exchange. Nov. 2022. URL: https://tex.stackexchange.com/a/666265 (visited on 12/03/2022) (cit. on p. 72).

[45] Qrrbrbirlbel. *Answer to "TikZ - 'Circled' arrow"*. TeX - LaTeX Stack Exchange. Jan. 2013. URL: https://tex.stackexchange.com/a/95263 (visited on 09/24/2022).

[46] Qrrbrbirlbel. *Answer to "TikZ - Four Colored Circle Split"*. June 2013. URL: https://tex.stackexchange.com/a/121767 (visited on 09/24/2022).

[47] Qrrbrbirlbel. *Answer to "TikZ / calendar: Set the height of a monthly calendar"*. Aug. 2022. URL: https://tex.stackexchange.com/a/653146 (visited on 09/24/2022).

[48] Qrrbrbirlbel. *Answer to "TikZ arrow tip is displaced"*. TeX - LaTeX Stack Exchange. Apr. 2013. URL: https://tex.stackexchange.com/a/111053 (visited on 04/02/2023).

[49] Qrrbrbirlbel. *Answer to "TikZ calendar and conditional tests"*. Oct. 2013. URL: https://tex.stackexchange.com/a/141027 (visited on 09/24/2022).

[50] Qrrbrbirlbel. *Answer to "TikZ: Define pattern with reference to external picture"*. Apr. 2013. URL: https://tex.stackexchange.com/a/107144 (visited on 09/24/2022).

[51] Qrrbrbirlbel. *Answer to "TikZ: How to place a coordinate at parabola-path-position?"* Nov. 2021. URL: https://tex.stackexchange.com/a/621012 (visited on 09/24/2022).

[52] somenxavier. *An oval surrounded a *long text* inside in TikZ [equivalent cover background of METAFUN]*. TeX - LaTeX Stack Exchange. Aug. 2022. URL: https://tex.stackexchange.com/q/649144 (visited on 09/24/2022).

[53] sro5h. *Achieve desired alignment of arrows in tikz-cd diagram*. TeX - LaTeX Stack Exchange. July 2022. URL: https://tex.stackexchange.com/q/652540 (visited on 02/19/2023).

[54] Andrew Stacey. *spath3 TikZ library*. original-date: 2014-05-26T12:08:12Z. Dec. 2022. URL: https://github.com/loopspace/spath3 (visited on 12/10/2022).

[55] Michał Szymankiewicz. *How to draw a mixing rule? #chemistry*. TeX - LaTeX Stack Exchange. Sept. 2022. URL: https://tex.stackexchange.com/q/657432 (visited on 10/23/2022).

[56] uulinux. *Is there a package to implement this style of "Register diagrams with field descriptions"*. TeX - LaTeX Stack Exchange. Oct. 2021. URL: https://tex.stackexchange.com/q/618047 (visited on 12/03/2022) (cit. on p. 72).

[57] Sašo Živanović. *Memoize*. original-date: 2020-05-19T09:58:52Z. Oct. 2023. URL: https://github.com/sasozivanovic/memoize (visited on 11/05/2023).