

The wheelchart package

Draw wheelcharts with TikZ

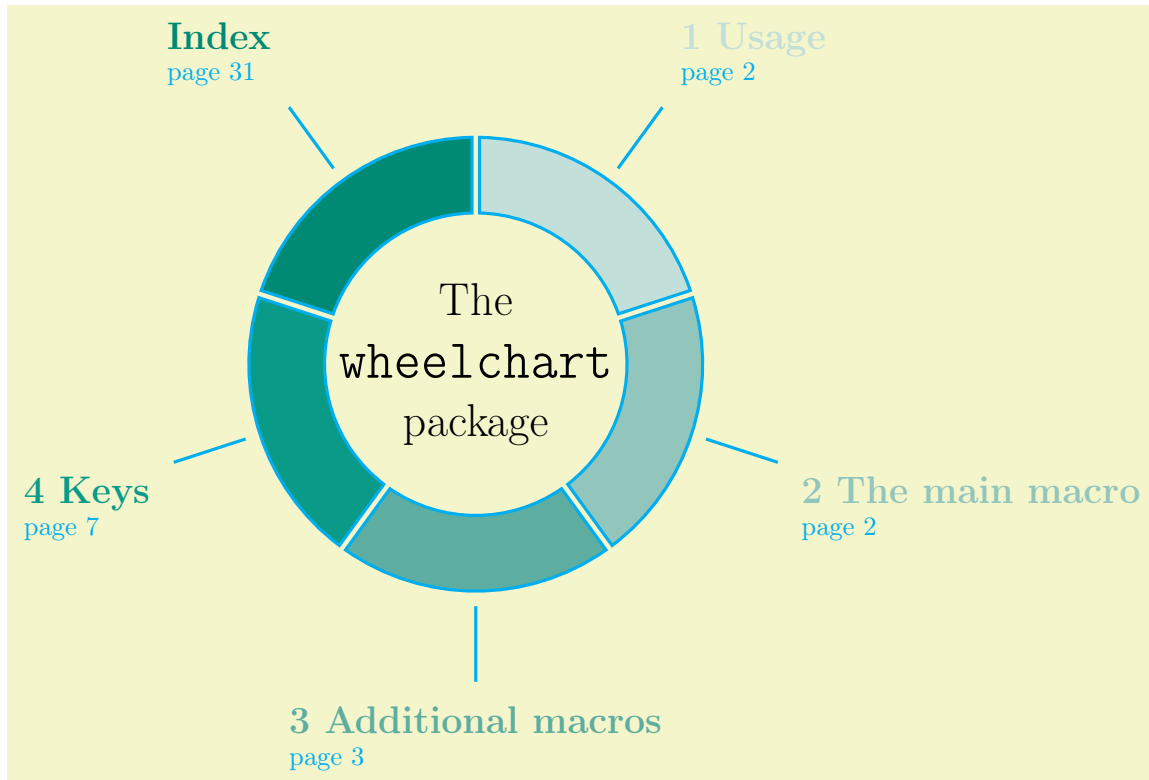
Matthias Floré

Version 1.0 (2022/09/11)

Abstract

This package is based on the package `tikz` (see [5]) and can be used to draw wheelcharts with TikZ. It provides several options to customize the wheelcharts. Other tools for creating wheelcharts or pie charts can be found in [2], [1], [4], [3] and [6].

Contents



```

\begin{tikzpicture}
\wheelchart[
  data={%
\pgfmathsetmacro{WCcolornumber}{(WCcount/WCtotalcount)*100}%
\ifdefempty{WCvarB}{\gdef\WChyperlink{section*.3}}{\gdef\WChyperlink{section.\WCcount}}%
\hyperlink{\WChyperlink}{%
\textbf{\textcolor{PineGreen!\WCcolornumber}\Large\ifdefempty{WCvarB}{\WCvarB}{\WCvarA}}}\|}%
\hyperlink{\WChyperlink}{\textcolor{Cyan}{page \WCvarC}}%
},
  gap,
  lines,
  lines style={Cyan,very thick},
  middle={\LARGE The\| [10pt] huge\texttt{wheelchart}\| [10pt] \LARGE package},
  slices style={
    /utils/exec={\pgfmathsetmacro{WCcolornumber}{(WCcount/WCtotalcount)*100}},
    PineGreen!\WCcolornumber,
    draw=Cyan,
    very thick
  },
  value=1
]{\WCtableofcontents}
\end{tikzpicture}

```

1 Usage

The package `wheelchart` can be used by putting the following in the preamble.

```
\usepackage{wheelchart}
```

The package `wheelchart` loads the package `tikz` and the TikZ library `calc`.

Many examples in this manual use colors which can be defined by giving `dvipsnames` as an option to `\documentclass`.

2 The main macro

```
\wheelchart[options]{wheelchart data}
```

This command can be placed inside a `tikzpicture` environment. It draws a wheelchart with *wheelchart data*. The *wheelchart data* is a comma separated list. Each item in this list corresponds to one slice of the wheelchart and consists of data separated by a `/`. The precise syntax of the *wheelchart data* will be explained below. The *options* can be given with the keys described in Section 4.

```
\exampleforthismanual
```

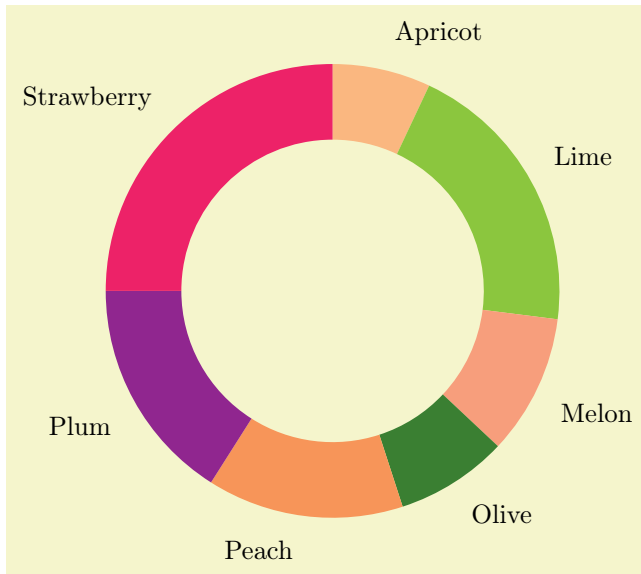
To simplify the creation of examples in this manual, we define the *wheelchart data* below.

```

\gdef\exampleforthismanual{%
14/Apricot/Apricot/{A, B, C, E, K}/{north east lines}/0/0/Gray,
40/LimeGreen/Lime/{B, C}/grid/0/15/Black,
20/Melon/Melon/{A, C}/0.5/0/none,
16/OliveGreen/Olive/{A, B, E, K}/dots/0/0/none,
28/Peach/Peach/{A, B, C, E, K}/{fivepointed stars}/0/0/Lavender,
32/Plum/Plum/{A, B, C, E, K}/bricks/0/{-15}/none,
50/WildStrawberry/Strawberry/{B, C, E, K}/1/0/DarkOrchid%
}

```

The default wheelchart with these data is shown below.

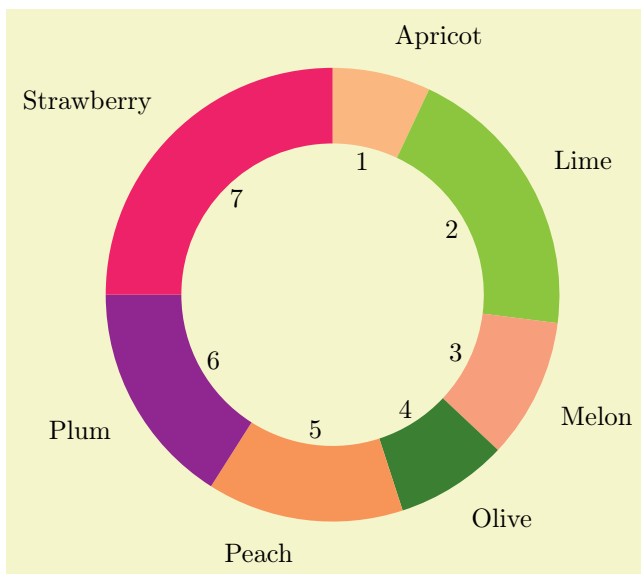


```
\begin{tikzpicture}
\wheelchart{\exampleforthismanual}
\end{tikzpicture}
```

3 Additional macros

`\WCcount`

This macro gives the current number of the slice in the `\wheelchart data`.



```
\begin{tikzpicture}
\wheelchart[
  inner data=\WCcount
]{\exampleforthismanual}
\end{tikzpicture}
```

`\WCdataangle`

This macro stores the sum of the value of the key `data angle shift` (taking into account the key `counterclockwise`) and the macro `\WCmidangle` modulo 360.

`\WCmidangle`

This macro gives the angle in degrees modulo 360 of the middle of the current slice.



```

\begin{tikzpicture}
\wheelchart[
  data angle shift=\WCvarG,
  data style={
    rotate=\WCdataangle,
    draw=Magenta,
    fill=GreenYellow,
    anchor=west,
    text=Gray
  }
]{\exampleforthismanual}
\wheelchart[
  data={},
  inner data={%
    \textbackslash WCmidangle%
  },
  inner data style={
    rotate=\WCmidangle,
    font=\ttfamily
  },
  slices style={fill=none}
]{\exampleforthismanual}
\end{tikzpicture}

```

\WCperc

This macro displays `\WCpercentage` rounded up to the number of decimals determined by the key `perc precision` followed by a % symbol.

If the package `siunitx` is loaded then the following code is used. The package `siunitx` can be loaded before or after the package `wheelchart`.

```

\qty[round-mode=places,
  round-precision=\pgfkeysvalueof{/wheelchart/perc precision}]{\WCpercentage}{\percent}

```

If the package `siunitx` is not loaded then the following code is used.

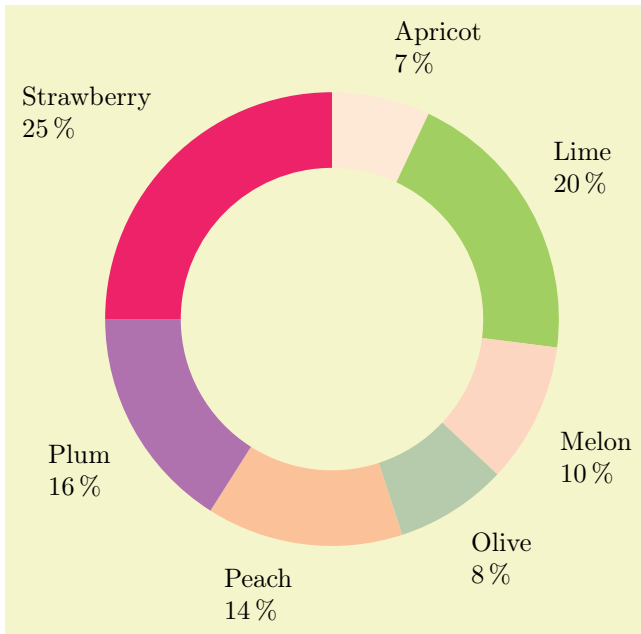
```

\WCpercentagerounded\,\%

```

\WCpercentage

This macro gives the percentage of the current slice where the total is computed with the values of the key `value`. Note that rounding errors can occur.



```

\begin{tikzpicture}
\wheelchart[
  data={\WCvarC||\WCperc},
  slices style={
    /utils/exec=\pgfmathsetmacro{\WCcolornumber}{4*\WCpercentage},
    \WCvarB!\WCcolornumber
  }
]{\exampleforthismanual}
\end{tikzpicture}

```

\WCpercentagerounded

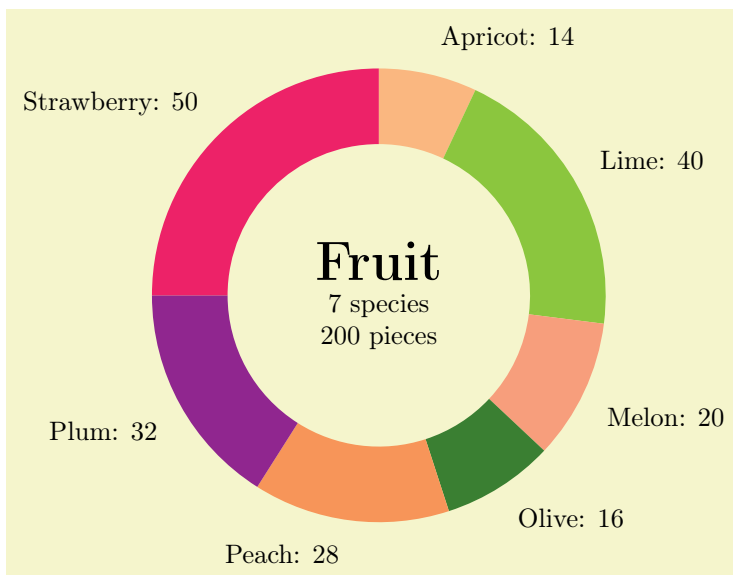
This macro displays \WCpercentage rounded up to the number of decimals determined by the key perc precision.

\WCtotalcount

This macro gives the total number of slices.

\WCtotalnum

This macro gives the sum of all values of the key value.



```

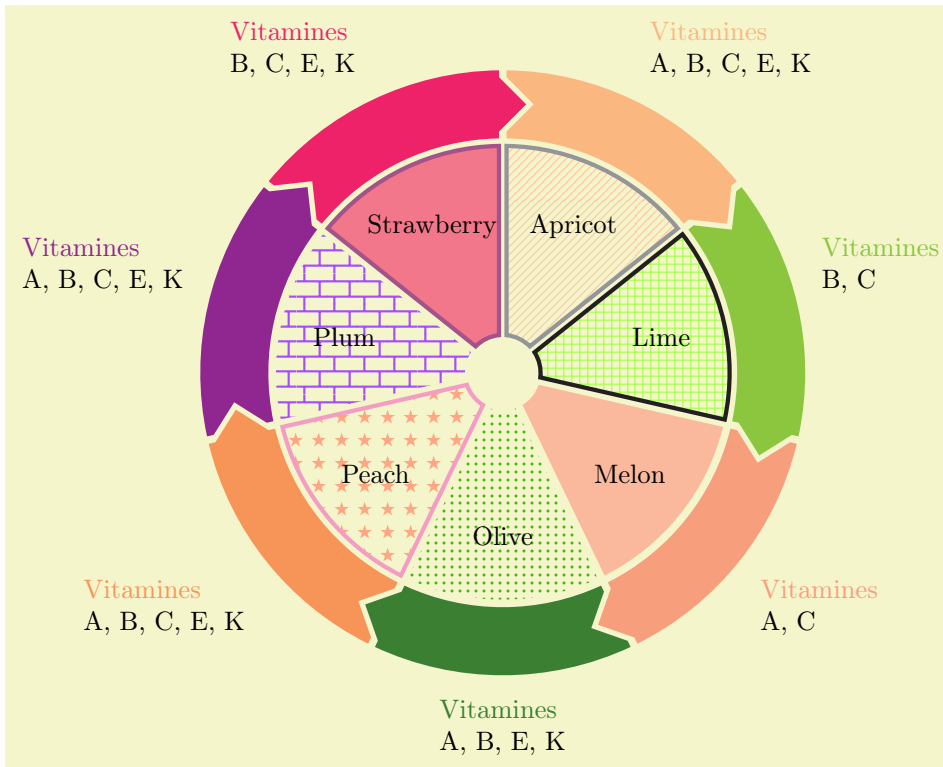
\begin{tikzpicture}
\wheelchart[
  data={\WCvarC: \WCvarA},
  middle={%
    \textbf{\huge Fruit}||%
    \WCtotalcount{} species||%
    \pgfmathprintnumber{\WCtotalnum}
    pieces%
  }
]{\exampleforthismanual}
\end{tikzpicture}

```

\WCvarA
\WCvarB
\WCvarC
\WCvarD
\WCvarE
\WCvarF
\WCvarG
\WCvarH
\WCvarI
\WCvarJ
\WCvarK
\WCvarL
\WCvarM
\WCvarN
\WCvarO
\WCvarP
\WCvarQ
\WCvarR
\WCvarS
\WCvarT
\WCvarU
\WCvarV
\WCvarW
\WCvarX
\WCvarY
\WCvarZ

The *⟨wheelchart data⟩* in the command `\wheelchart` is a comma separated list. Each item in this list corresponds to one slice of the wheelchart and consists of data separated by a `/`. These individual data are interpreted as `\WCvarA/\WCvarB/\WCvarC/...` and can be accessed within the *⟨options⟩* of the command `\wheelchart` by the macros `\WCvarA` till `\WCvarZ` except within the keys `at`, `caption`, `caption left`, `caption left style`, `caption style`, `contour`, `counterclockwise`, `expand list`, `middle fill`, `name`, `start angle`, `start half`, `title`, `title left`, `title left style`, `title style`, `total angle` and `total count`. Thus up to 26 data can be given to each slice of the wheelchart.

Initially, only `\WCvarA`, `\WCvarB` and `\WCvarC` are used for `value=\WCvarA`, `slices style=\WCvarB` and `data=\WCvarC`.



```

\usetikzlibrary {patterns}
\begin{tikzpicture}
\wheelchart[
  data={},
  gap,
  radius={0.5}{3},
  slices style={\WCvarB!70,draw=\WCvarH,ultra thick,pattern=\WCvarE,pattern color=\WCvarB!70},
  value=1,
  wheel data=\WCvarC,
  %wheel data style={shift={(\WCmidangle:0.5)}},
  %wheel data pos=0.5
][\exampleforthismanual]
\wheelchart[
  data={\textcolor{\WCvarB}{Vitamines}}|\WCvarD},
  gap,
  radius={3.1}{4},
  slices arrow={1}{0.2},
  value=1
][\exampleforthismanual]
\end{tikzpicture}

```

4 Keys

The keys in this Section can be given as *options* to the command `\wheelchart`.

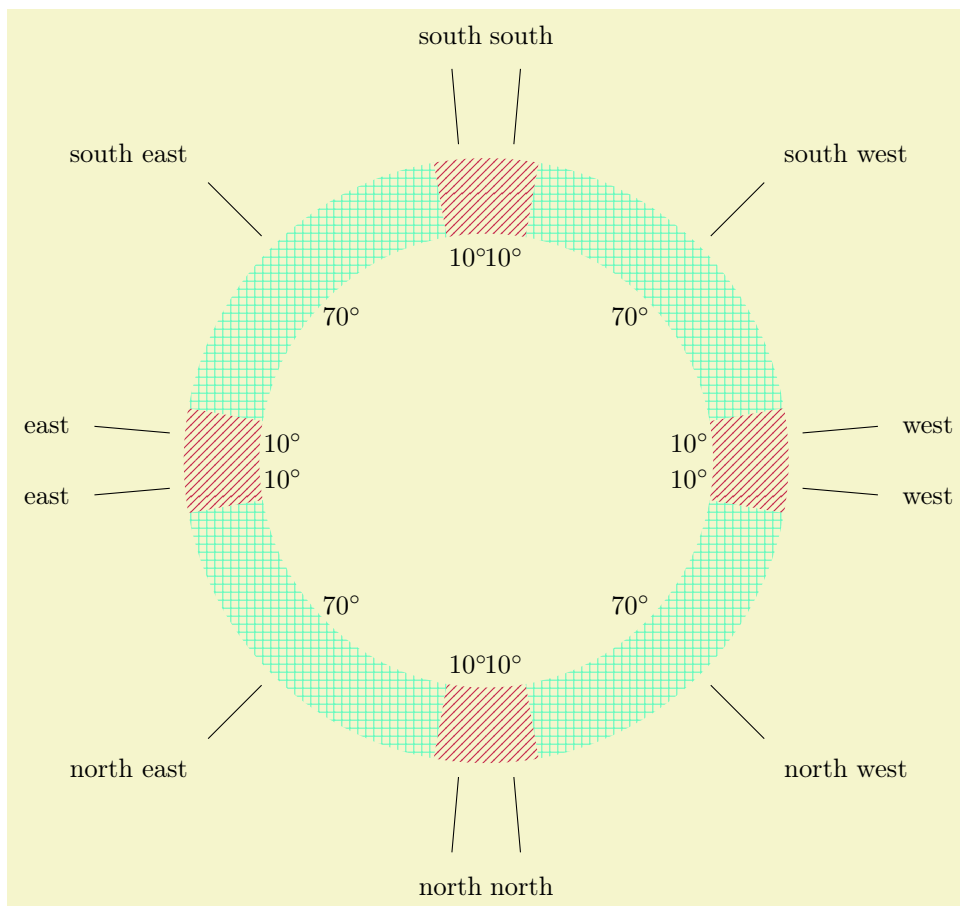
`/wheelchart/anchor xsep={angle}` (no default, initially 5)

`/wheelchart/anchor ysep={angle}` (no default, initially 5)

These keys determine the default anchor of the key `data` in the case that `lines ext=0`.

<code>\WCdataangle</code>	Anchor of the key data in the case that <code>lines ext=0</code>
0	west
90	south
180	east
270	north
For other angles not in $\{0, 90, 180, 270\}$:	
<code>[0, anchor ysep]</code>	west
<code>]anchor ysep, 90 - anchor xsep[</code>	south west
<code>[90 - anchor xsep, 90 + anchor xsep]</code>	south
<code>]90 + anchor xsep, 180 - anchor ysep[</code>	south east
<code>[180 - anchor ysep, 180 + anchor ysep]</code>	east
<code>]180 + anchor ysep, 270 - anchor xsep[</code>	north east
<code>[270 - anchor xsep, 270 + anchor xsep]</code>	north
<code>]270 + anchor xsep, 360 - anchor ysep[</code>	north west
<code>[360 - anchor ysep, 360]</code>	west

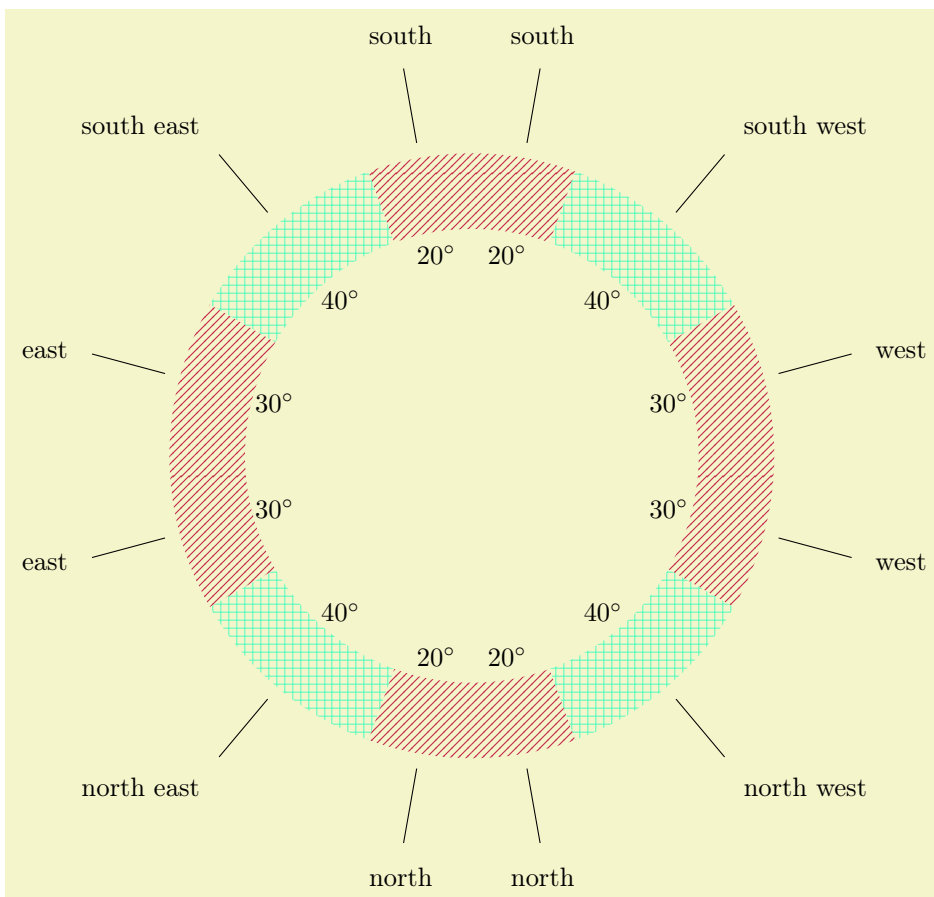
Table 1: Anchor of the key data in the case that `lines ext=0`.




```

\usetikzlibrary {patterns}
\begin{tikzpicture}
\wheelchart[
  inner data={\WCvarA~{\circ}$},
  %inner data style={shift={(\WCmidangle:{-0.1})}},
  inner data sep=0.3,
  lines,
  radius={3}{4},
  slices style={pattern=\WCvarD,pattern color=\WCvarB!70},
]{}%
  10/Maroon/south/{north east lines},
  70/TealBlue/{south west}/grid,
  10/Maroon/west/{north east lines},
  10/Maroon/west/{north east lines},
  70/TealBlue/{north west}/grid,
  10/Maroon/north/{north east lines},
  10/Maroon/north/{north east lines},
  70/TealBlue/{north east}/grid,
  10/Maroon/east/{north east lines},
  10/Maroon/east/{north east lines},
  70/TealBlue/{south east}/grid,
  10/Maroon/south/{north east lines}%
}
\end{tikzpicture}

```

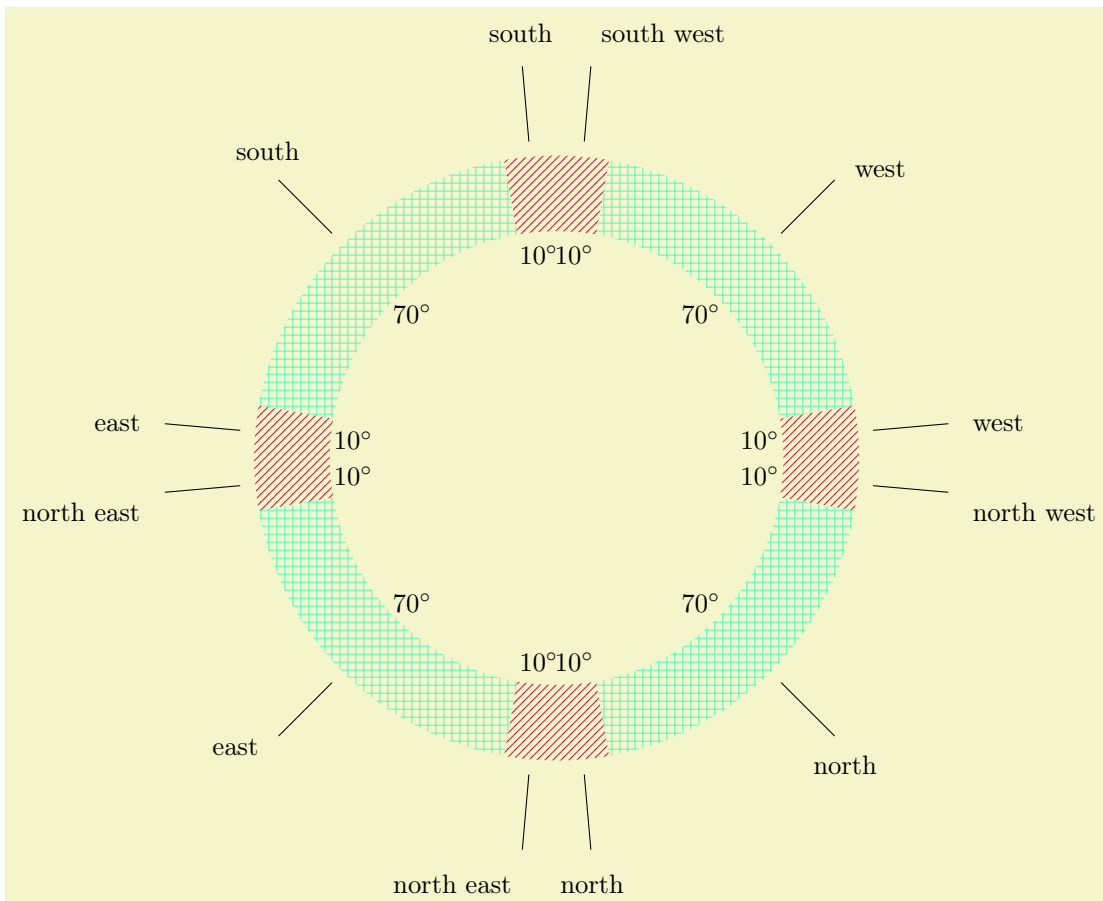


```

\usetikzlibrary {patterns}
\begin{tikzpicture}
\wheelchart[
  anchor xsep=10,
  anchor ysep=15,
  inner data={\WCvarA~{\circ}$},
  %inner data style={shift={(\WCmidangle:{-0.1})}},
  inner data sep=0.3,
  lines,
  radius={3}{4},
  slices style={pattern=\WCvarD,pattern color=\WCvarB!70},
]{%
  20/Maroon/south/{north east lines},
  40/TealBlue/{south west}/grid,
  30/Maroon/west/{north east lines},
  30/Maroon/west/{north east lines},
  40/TealBlue/{north west}/grid,
  20/Maroon/north/{north east lines},
  20/Maroon/north/{north east lines},
  40/TealBlue/{north east}/grid,
  30/Maroon/east/{north east lines},
  30/Maroon/east/{north east lines},
  40/TealBlue/{south east}/grid,
  20/Maroon/south/{north east lines}%
}
\end{tikzpicture}

```

The anchor of the key data can also be specified manually by using the key `data style`.



```

\usetikzlibrary {patterns}
\begin{tikzpicture}
\wheelchart[
  data style={anchor=\WCvarC},
  inner data={\WCvarA~{\circ}\WCvarB},
  %inner data style={shift={(\WCmidangle:-0.1)}},
  inner data sep=0.3,
  lines,
  radius={3}{4},
  slices style={pattern=\WCvarD,pattern color=\WCvarB!70},
]{%
  10/Maroon/{south west}/{north east lines},
  70/TealBlue/west/grid,
  10/Maroon/west/{north east lines},
  10/Maroon/{north west}/{north east lines},
  70/TealBlue/north/grid,
  10/Maroon/north/{north east lines},
  10/Maroon/{north east}/{north east lines},
  70/TealBlue/east/grid,
  10/Maroon/{north east}/{north east lines},
  10/Maroon/east/{north east lines},
  70/TealBlue/south/grid,
  10/Maroon/south/{north east lines}%
}
\end{tikzpicture}

```

`/wheelchart/at={\langle point \rangle}` (no default, initially (0,0))

This key defines the center of the wheelchart.

`/wheelchart/caption={\langle text \rangle}` (no default)

This key contains the $\langle text \rangle$ which will be placed below the wheelchart. The $\langle text \rangle$ is placed in a node. The x coordinate of this node is the x coordinate of the center of the wheelchart, which is defined by the key `at`. In general, this is *not* the same as the x coordinate of the center of the local bounding box around the wheelchart. The y coordinate of this node is 0.5 below the south of the local bounding box around the wheelchart. The style of this node is given as follows. First, the options `anchor=north,align=center` are given. Thereafter, the style of the key `caption style` is added.

`/wheelchart/caption left={\langle text \rangle}` (no default)

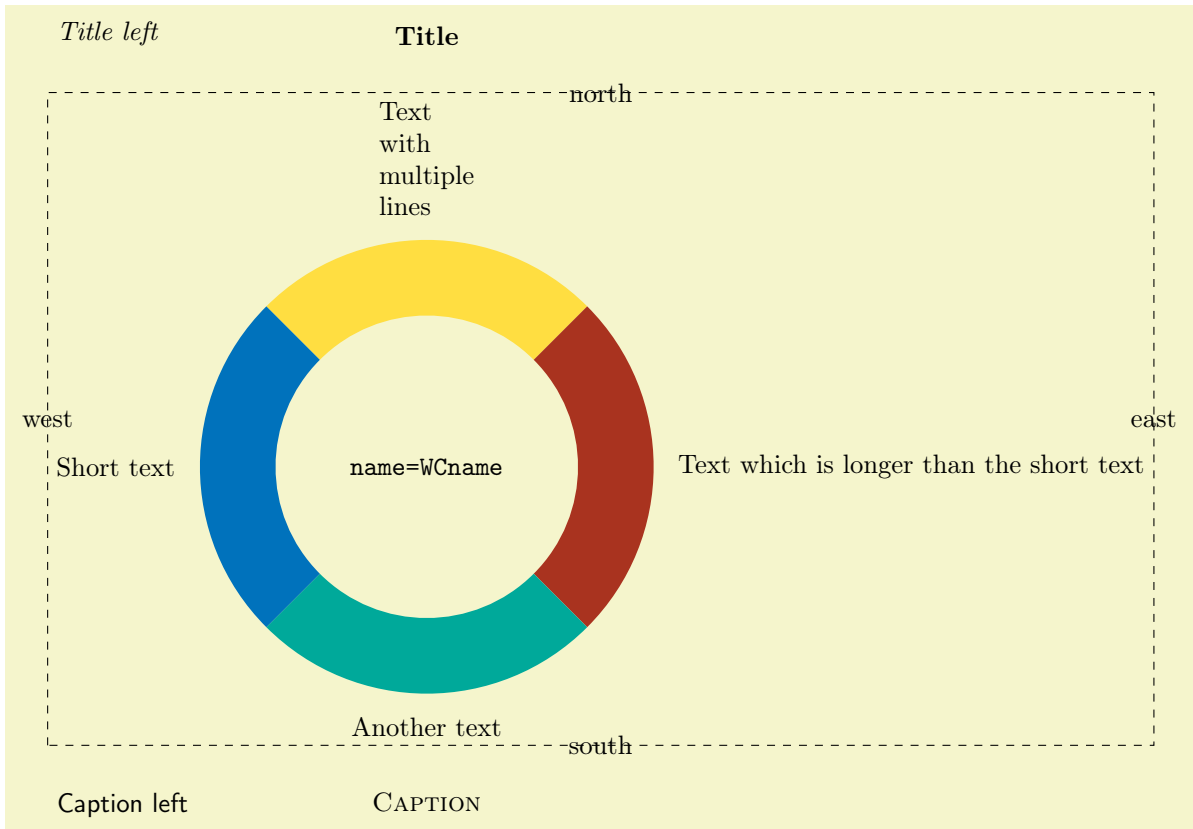
This key contains the $\langle text \rangle$ which will be placed below left of the wheelchart. The $\langle text \rangle$ is placed in a node. This node is placed 0.5 below the south west of the local bounding box around the wheelchart. The style of this node is given as follows. First, the options `anchor=north west,align=left` are given. Thereafter, the style of the key `caption left style` is added.

`/wheelchart/caption left style={\langle options \rangle}` (style, no default, initially empty)

This key accepts a list of keys which will be applied to the node where the contents of the key `caption left` is placed.

`/wheelchart/caption style={\langle options \rangle}` (style, no default, initially empty)

This key accepts a list of keys which will be applied to the node where the contents of the key `caption` is placed.



```

\begin{tikzpicture}
\wheelchart[
  at={(5,2)},
  caption=Caption,
  caption style={font=\scshape},
  caption left={Caption left},
  caption left style={font=\sfamily},
  middle={\texttt{name=WCname}},
  name=WCname,
  start half,
  title=Title,
  title style={font=\bfseries},
  title left={Title left},
  title left style={font=\em}
]{%
  1/Goldenrod/{Text\with\multiple\lines},
  1/Mahogany/{Text which is longer than the short text},
  1/JungleGreen/{Another text},
  1/RoyalBlue/{Short text}%
}
\draw[dashed] (WCname.south west) rectangle (WCname.north east);
\foreach\pos in {north,east,south,west}{
  \node at (WCname.\pos) {\pos};
}
\end{tikzpicture}

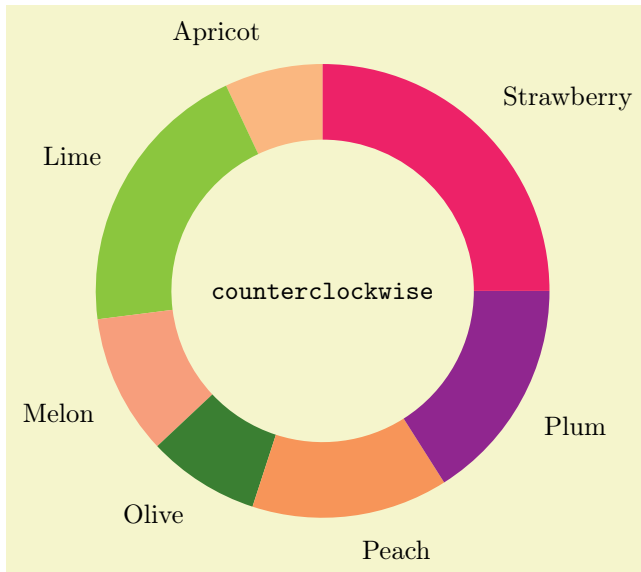
```

`/wheelchart/contour={\langle options \rangle}` (style, no default, initially empty)

If this key is set then a contour with the style determined by this key will be drawn around the wheelchart.

`/wheelchart/counterclockwise={\langle boolean \rangle}` (default true, initially false)

If true, the wheelchart will be drawn counterclockwise instead of clockwise.



```
\begin{tikzpicture}
\wheelchart[
  counterclockwise,
  middle=counterclockwise,
  middle style={font=\ttfamily}
]{\exampleforthismanual}
\end{tikzpicture}
```

`/wheelchart/data={\langle text \rangle}` (no default, initially `\WCvarC`)

This key contains the `\langle text \rangle` which will be placed at the outside of each slice of the wheelchart. This can be suppressed by using `data={}`. The `\langle text \rangle` is placed in a node. The style of this node is given as follows. First, the anchor is set following Table 1 and Table 2. Then the option `align=left` is added. Thereafter, the style of the key `data style` is added.

`/wheelchart/data angle shift={\langle angle \rangle}` (no default, initially 0)

The contents of the key `data` is placed at the angle `\WCdataangle`, which is the sum of the value of the key `data angle shift` in degrees (taking into account the key `counterclockwise`) and the macro `\WCmidangle` modulo 360.

`/wheelchart/data sep={\langle value \rangle}` (no default, initially 0.2)

If `lines=0`, this key defines the distance between the wheelchart and the contents of the key `data`. If `lines > 0`, this key defines the distance between the end of the lines and the contents of the key `data`.

`/wheelchart/data style={\langle options \rangle}` (style, no default, initially empty)

This key accepts a list of keys which will be applied to the node where the contents of the key `data` is placed.

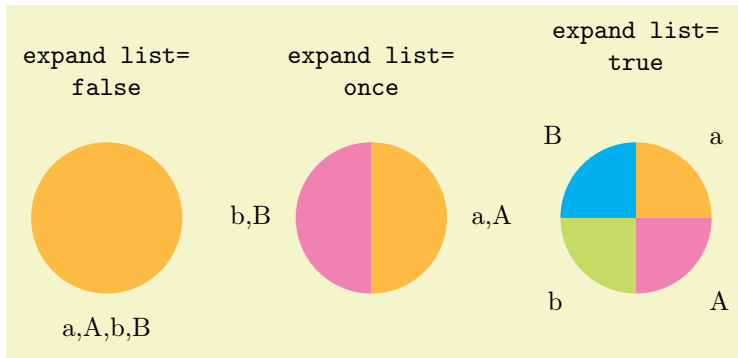
`/wheelchart/expand list=false|once|true` (no default, initially `once`)

`false` In this case, the `\langle wheelchart data \rangle` of the command `\wheelchart` will not be expanded.

`once` In this case, the `\langle wheelchart data \rangle` of the command `\wheelchart` will be expanded once.

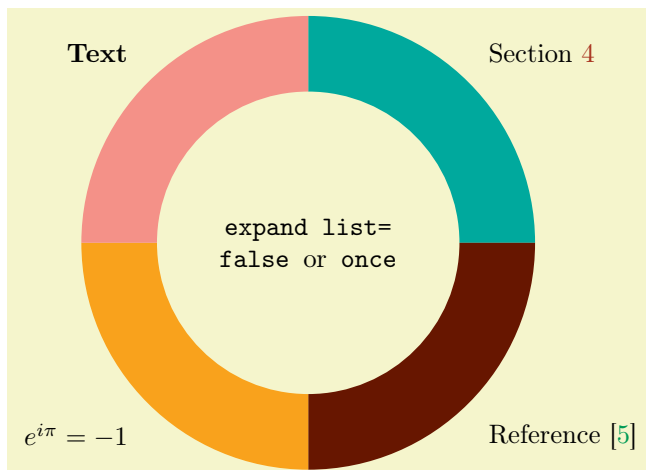
`true` In this case, the `\langle wheelchart data \rangle` of the command `\wheelchart` will be fully expanded.

The following example illustrates the difference between the possible values of the key `expand list`.



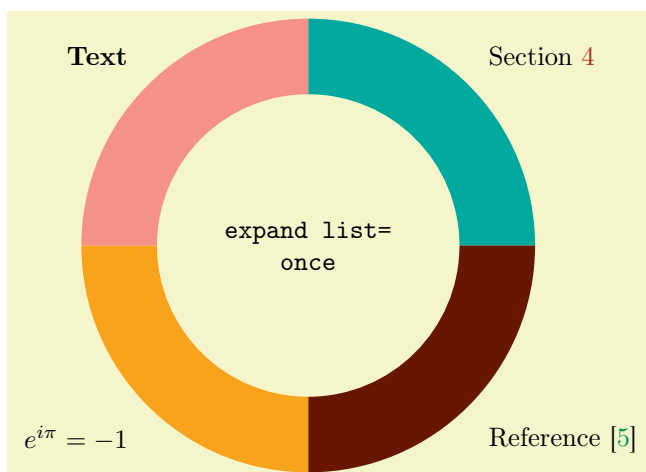
```
\usepackage {listofitems}
\readlist*\WCcolors{
  Dandelion,CarnationPink,
  SpringGreen,ProcessBlue
}
\begin{tikzpicture}
\def\WclistA{a,A}
\def\WclistB{b,B}
\def\WCdata{\WclistA,\WclistB}
\foreach\expandlist [count=\n] in
{false,once,true}{
\wheelchart[
  at={{(3.5*\n,0)},
  data=\WCvarA,
  expand list=\expandlist,
  radius={0}{1},
  slices style={\WCcolors[\WCcount]},
  title={expand list=|\expandlist},
  title style={font=\ttfamily},
  value=1
] {\WCdata}
}
\end{tikzpicture}
```

The initial setting `expand list=once` works in most situations, even when commands such as `\ref`, `\cite` and `\textbf` are used such as in the example below.



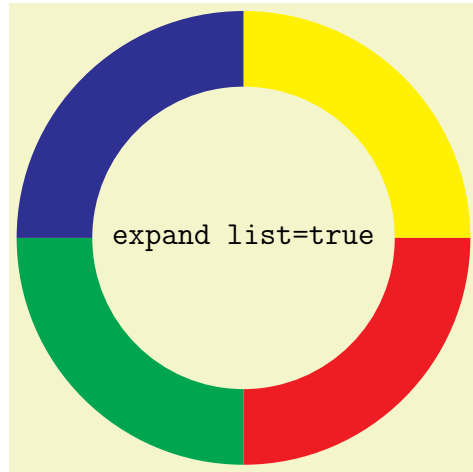
```
\begin{tikzpicture}
\wheelchart[
  %expand list=false,%false also works
  %expand list=true,%true doesn't work
  middle={expand list=|\false %
    {\normalfont or} once},
  middle style={font=\ttfamily}
] {%
  1/Emerald/{Section \ref{Keys}},
  1/Sepia/{Reference \cite{TtTaPGFp}},
  1/YellowOrange/{$e^{i\pi}=-1$},
  1/Salmon/{\textbf{Text}}%
}
\end{tikzpicture}
```

In the following example, the `\wheelchart data` from the previous example is stored in a macro. In this case, we have to use the initial setting `expand list=once`.



```
\begin{tikzpicture}
\def\Wclist{%
  1/Emerald/{Section \ref{Keys}},
  1/Sepia/{Reference \cite{TtTaPGFp}},
  1/YellowOrange/{$e^{i\pi}=-1$},
  1/Salmon/{\textbf{Text}}%
}
\wheelchart[
  %expand list=false,
  %expand list=true,
  %false and true do not work
  middle={expand list=|\once},
  middle style={font=\ttfamily}
] {\Wclist}
\end{tikzpicture}
```

In the example below, we have to use `expand list=true`.

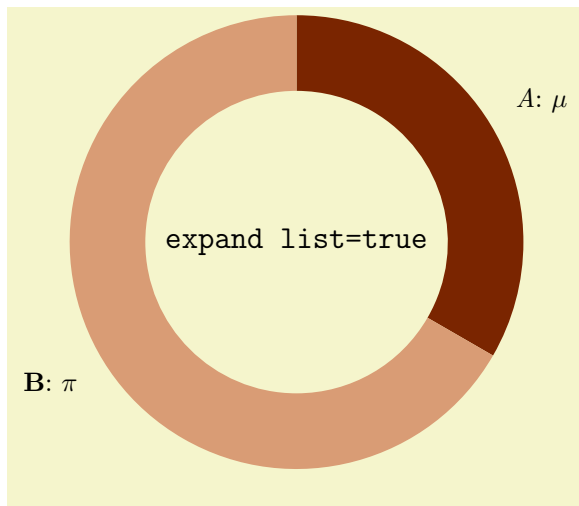


```

\begin{tikzpicture}
\def\WCcolorsA{Yellow,Red}
\def\WCcolorsB{Green,Blue}
\wheelchart[
  data={},
  expand list=true,%false and once
                    %do not work
  middle={expand list=true},
  middle style={font=\large\ttfamily},
  slices style={\WCvarA},
  value=1
]{\WCcolorsA,\WCcolorsB}
\end{tikzpicture}

```

In the example below, we have to use `expand list=true` and the command `\expandonce` from the package `etoolbox`.



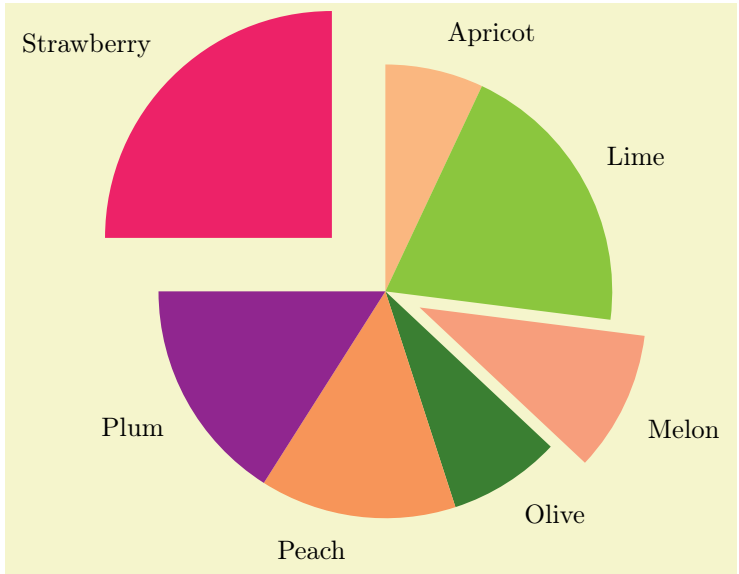
```

\usepackage {etoolbox}
\begin{tikzpicture}
\def\WCsliceA{1/Brown/{\emph{A}: $\mu$}}
\def\WCsliceAfinal{\expandonce\WCsliceA}
\def\WCsliceB{2/Tan/{\textbf{B}: $\pi$}}
\def\WCsliceBfinal{\expandonce\WCsliceB}
\wheelchart[
  expand list=true,%false and once
                    %do not work
  middle={expand list=true},
  middle style={font=\large\ttfamily},
]{\WCsliceAfinal,\WCsliceBfinal}
% \WCsliceA and \WCsliceB do not work
\end{tikzpicture}

```

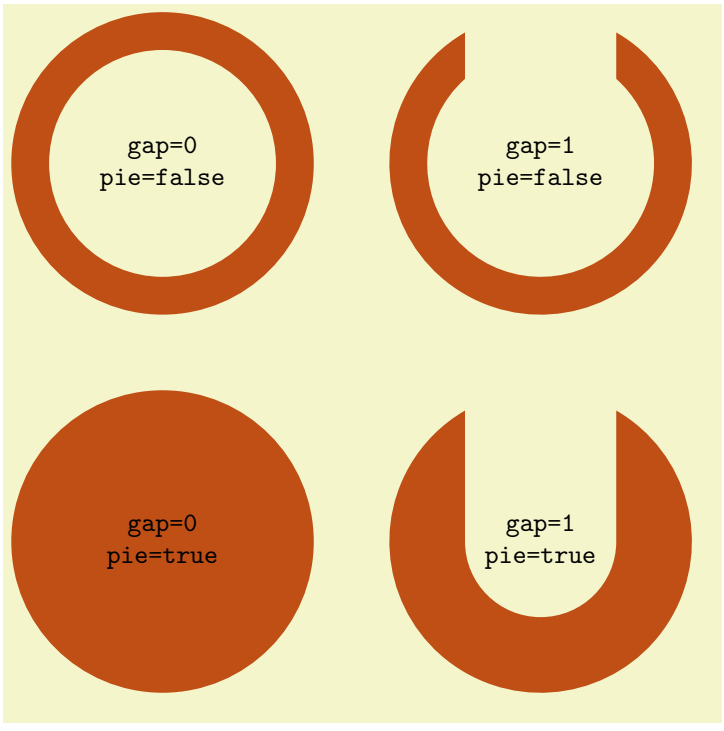
`/wheelchart/explode={\langle value \rangle}` (default 0.2, initially 0)

This key will shift the slices of the wheelchart with `\langle value \rangle` with respect to the center of the wheelchart.



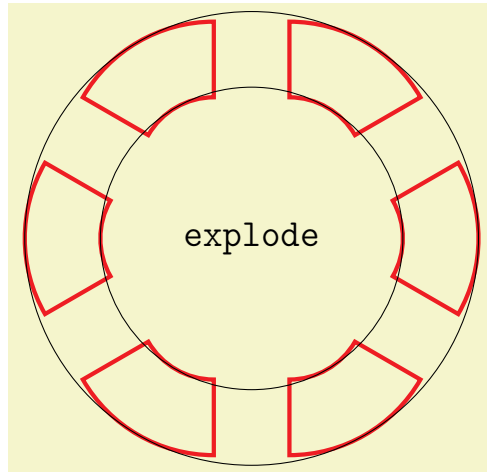
```
\begin{tikzpicture}
\wheelchart[
  explode=|WCvarF,
  pie
]{\exampleforthismanual}
\end{tikzpicture}
```

`/wheelchart/gap={⟨value⟩}` (default 0.05, initially 0)
 The *⟨value⟩* of this key defines half the distance between two slices of the wheelchart.
 The following example illustrates the behaviour of the key `gap` when a slice has 360 degrees.



```
\begin{tikzpicture}
\foreach\gap [count=\m] in {0,1}{
\foreach\pie [count=\n] in {false,true}{
\wheelchart[
  at={{5*\m},{-5*\n}},
  data={},
  gap=\gap,
  middle={gap=\gap|pie=\pie},
  middle style={font=\ttfamily},
  pie=\pie,
  radius={1.5}{2},
  slices style=Bittersweet,
  total count=1,
  value=1
]{}
}
}
\end{tikzpicture}
```

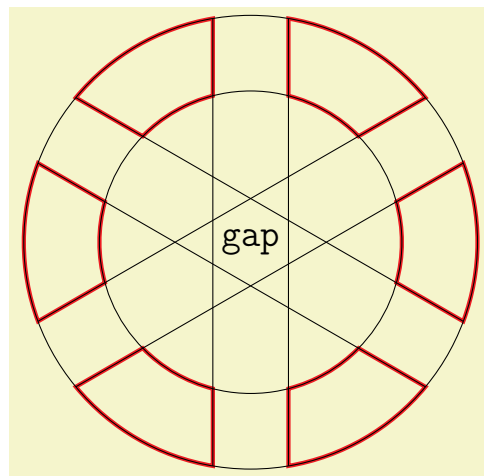
`/wheelchart/gap polar={⟨value⟩}` (default 1, initially 0)
 The *⟨value⟩* of this key defines half the polar gap in degrees between two slices of the wheelchart.
 Note the difference between the keys `explode`, `gap` and `gap polar`. This is illustrated in the examples below.



```

\begin{tikzpicture}
\wheelchart[
  data={},
  explode=1,
  middle={\Large\texttt{explode}},
  radius={1}{2},
  slices style={
    draw=Red,
    fill=none,
    ultra thick
  },
  total count=6,
  value=1
]{}
\draw (0,0) circle[radius=2];
\draw (0,0) circle[radius=3];
\end{tikzpicture}

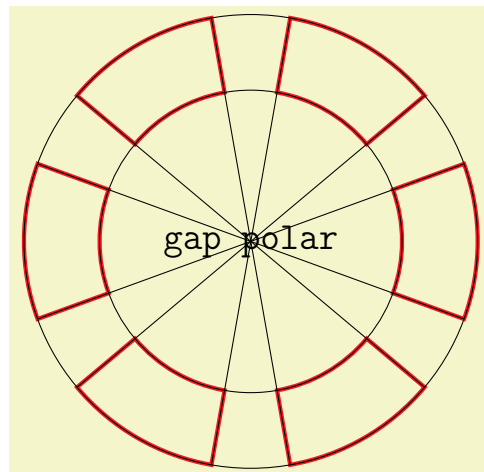
```



```

\begin{tikzpicture}
\wheelchart[
  data={},
  gap=0.5,
  middle={\Large\texttt{gap}},
  slices style={
    draw=Red,
    fill=none,
    ultra thick
  },
  total count=6,
  value=1
]{}
\draw (0,0) circle[radius=2];
\draw (0,0) circle[radius=3];
\foreach \a in {0,60,120}{
\foreach \x in {-0.5,0.5}{
\draw[rotate=\a] (\x,{sqrt(3^2-0.5^2)}--
(\x,-sqrt(3^2-0.5^2));
}
}
\end{tikzpicture}

```



```

\begin{tikzpicture}
\wheelchart[
  data={},
  gap polar=10,
  middle={\Large\texttt{gap polar}},
  slices style={
    draw=Red,
    fill=none,
    ultra thick
  },
  total count=6,
  value=1
]{}
\draw (0,0) circle[radius=2];
\draw (0,0) circle[radius=3];
\foreach \a in {30,90,150}{
\foreach \t in {-10,10}{
\draw ({\t+\a}:3)--({\t+\a+180}:3);
}
}
\end{tikzpicture}

```

`/wheelchart/inner data={\langle text \rangle}`

(no default)

This key contains the $\langle text \rangle$ which will be placed at the inside of each slice of the wheelchart. The $\langle text \rangle$ is placed in a node. The style of this node is given as follows. First, the option `align=left` is given. Thereafter, the style of the key `inner data style` is added.

`/wheelchart/inner data sep={⟨value⟩}` (no default, initially 0.2)

This key defines the distance between the wheelchart and the contents of the key `inner data`.

`/wheelchart/inner data style={⟨options⟩}` (style, no default, initially empty)

This key accepts a list of keys which will be applied to the node where the contents of the key `inner data` is placed.

`/wheelchart/inner radius={⟨value⟩}` (no default, initially 2)

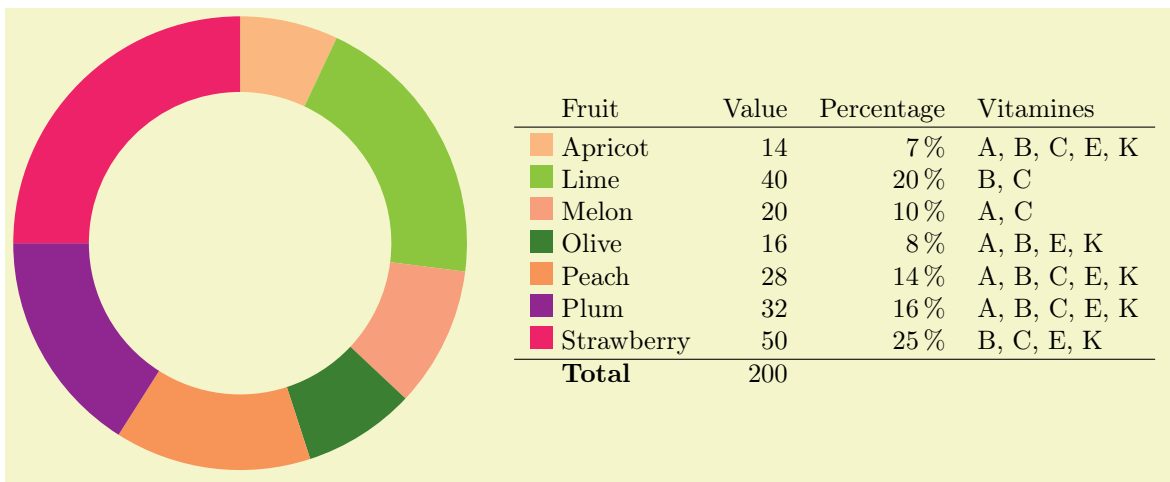
The `⟨value⟩` of this key defines the inner radius of the wheelchart.

`/wheelchart/legend={⟨code⟩}` (no default)

If this key is set then the `⟨code⟩` given to this key will be executed at the end of the command `\wheelchart`.

`/wheelchart/legend entry={⟨code⟩}` (no default)

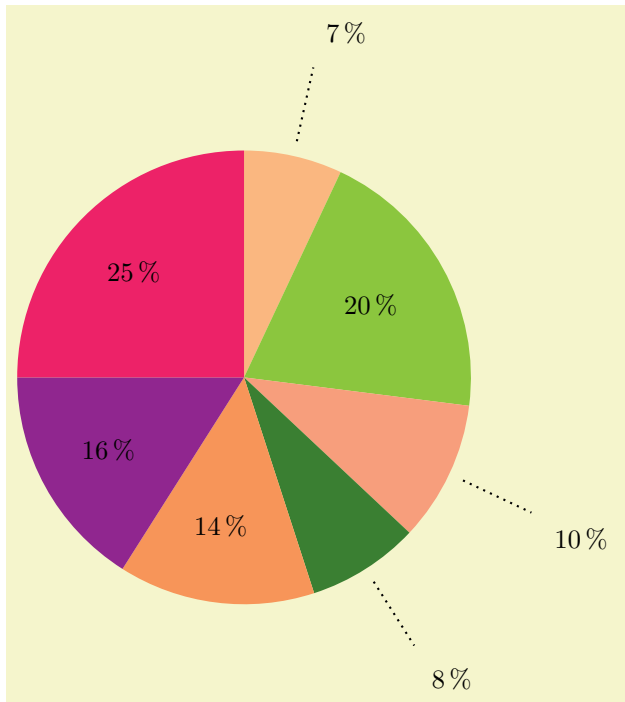
If this key is set then the `⟨code⟩` given to this key will be executed for each slice of the wheelchart.



```
\usepackage {etoolbox}
\begin{tikzpicture}
\def\WClegend{}
\def\WClegendrow#1#2#3#4#5{\tikz\fill[#1] (0,0) rectangle (0.3,0.3); & #2 & $#3$ & #4 & #5\\}
\wheelchart[
  data={},
  legend entry={
    |gappto\WClegend{\WClegendrow}
    \xappto\WClegend{\WCvarB}{\WCvarC}{\WCvarA}{\WCperc}{\WCvarD}}
  },
  legend={
    \node[anchor=west] at (3.5,0) {%
      \begin{tabular}{l@{ }lrrl}%
        & Fruit & Value & Percentage & Vitamines||\hline%
        ||[-10pt]%
        \WClegend\hline%
        & \textbf{Total} & & $\pgfmathprintnumber{\WCtotalnum}$ & ||%
        \end{tabular}%
      };
    }
] {\exampleforthismanual}
\end{tikzpicture}
```

`/wheelchart/lines={⟨value⟩}` (default 1, initially 0)

This key will draw lines of length `⟨value⟩` between the wheelchart and the contents of the key `data`.



```

\begin{tikzpicture}
\def\WCtest#1#2{%
  \ifdim \WCpercentage pt>10 pt%
    #1%
  \else%
    #2%
  \fi%
}
\wheelchart[
  data={\WCtest}{\WCperc}},
  lines={
    1-max(sign(\WCpercentage-10),0)
  },
  lines style={dotted,thick},
  pie,
  wheel data={\WCtest{\WCperc}}
]{\exampleforthismanual}
\end{tikzpicture}

```

`/wheelchart/lines ext={value}` (default 0.5, initially 0)

If the *value* of this key is > 0 then the lines between the wheelchart and the contents of the key `data` will be extended horizontally with a length defined by *value*.

`/wheelchart/lines ext bottom dir=left|right` (no default, initially right)

This key applies when $\backslash\text{WCdataangle} \in [270 - \text{lines ext dirsep}, 270 + \text{lines ext dirsep}]$. In this case, this key defines the direction in which the lines between the wheelchart and the contents of the key `data` will be extended horizontally and in this case, this key also determines the anchor of the key `data`.

left In this case, the lines between the wheelchart and the contents of the key `data` will be extended horizontally to the left and the anchor of the key `data` is the value of the key `lines ext left anchor`.

right In this case, the lines between the wheelchart and the contents of the key `data` will be extended horizontally to the right and the anchor of the key `data` is the value of the key `lines ext right anchor`.

`/wheelchart/lines ext dirsep={angle}` (no default, initially 0)

This key determines half the angle in degrees of the segment to which the keys `lines ext bottom dir` and `lines ext top dir` apply.

`/wheelchart/lines ext fixed={boolean}` (default true, initially false)

If true, all lines between the wheelchart and the contents of the key `data` will be extended horizontally till the same *x* coordinate at the left and till the same *x* coordinate at the right.

`/wheelchart/lines ext left anchor={anchor}` (no default, initially mid east)

This key defines the anchor of the key `data` when the lines between the wheelchart and the contents of the key `data` are extended horizontally to the left.

`/wheelchart/lines ext right anchor={anchor}` (no default, initially mid west)

This key defines the anchor of the key `data` when the lines between the wheelchart and the contents of the key `data` are extended horizontally to the right.

`/wheelchart/lines ext top dir=left|right` (no default, initially right)

This key applies when $\backslash\text{WCdataangle} \in [90 - \text{lines ext dirsep}, 90 + \text{lines ext dirsep}]$. In this case, this key defines the direction in which the lines between the wheelchart and the contents of the key `data` will be extended horizontally and in this case, this key also determines the anchor of the key `data`.

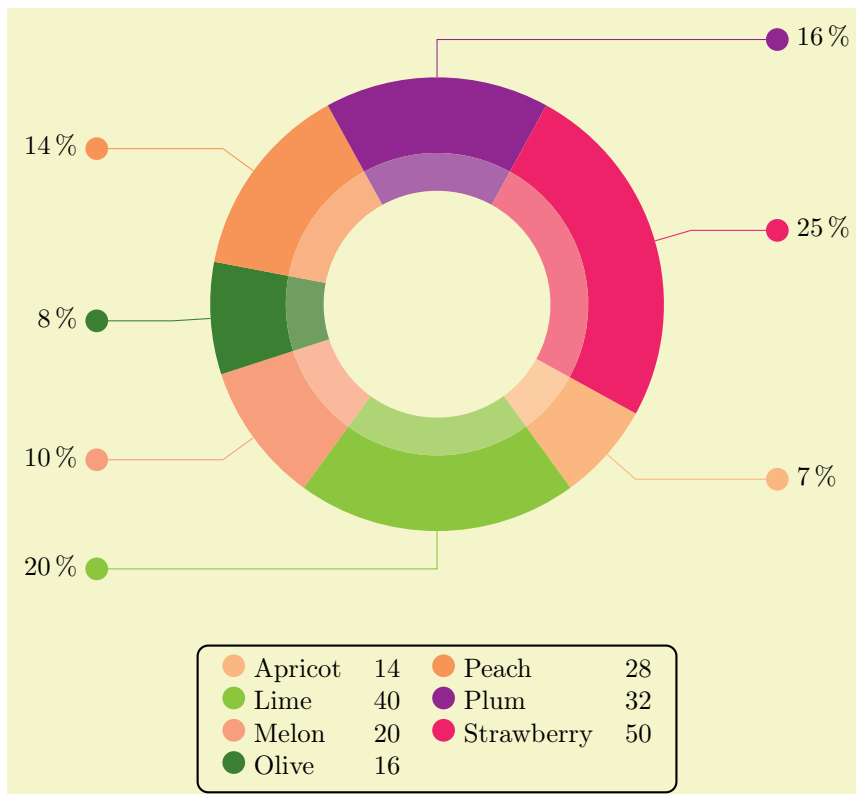
left In this case, the lines between the wheelchart and the contents of the key data will be extended horizontally to the left and the anchor of the key data is the value of the key `lines ext left anchor`.

right In this case, the lines between the wheelchart and the contents of the key data will be extended horizontally to the right and the anchor of the key data is the value of the key `lines ext right anchor`.

<code>\WCdataangle</code>	Anchor of the key data in the case that <code>lines ext > 0</code>
<code>[0,90 - lines ext dirsep[</code> <code>[90 - lines ext dirsep,90 + lines ext dirsep]</code>	value of the key <code>lines ext right anchor</code> value of the key <code>lines ext left anchor</code> if <code>lines ext top dir=left</code> value of the key <code>lines ext right anchor</code> if <code>lines ext top dir=right</code>
<code>]90 + lines ext dirsep,270 - lines ext dirsep[</code> <code>[270 - lines ext dirsep,270 + lines ext dirsep]</code>	value of the key <code>lines ext left anchor</code> value of the key <code>lines ext left anchor</code> if <code>lines ext bottom dir=left</code> value of the key <code>lines ext right anchor</code> if <code>lines ext bottom dir=right</code>
<code>]270 + lines ext dirsep,360[</code>	value of the key <code>lines ext right anchor</code>

Table 2: Anchor of the key data in the case that `lines ext > 0`.

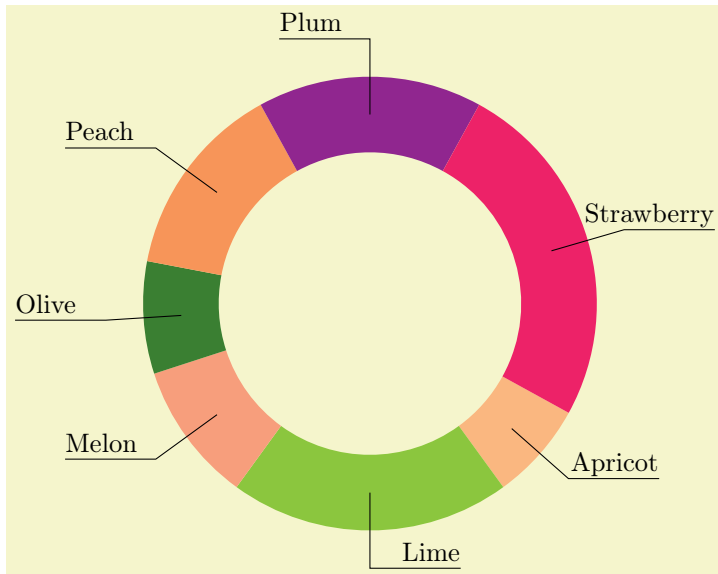
The number of columns in the legend in the example below can be specified with `\WClegendcolumns`.



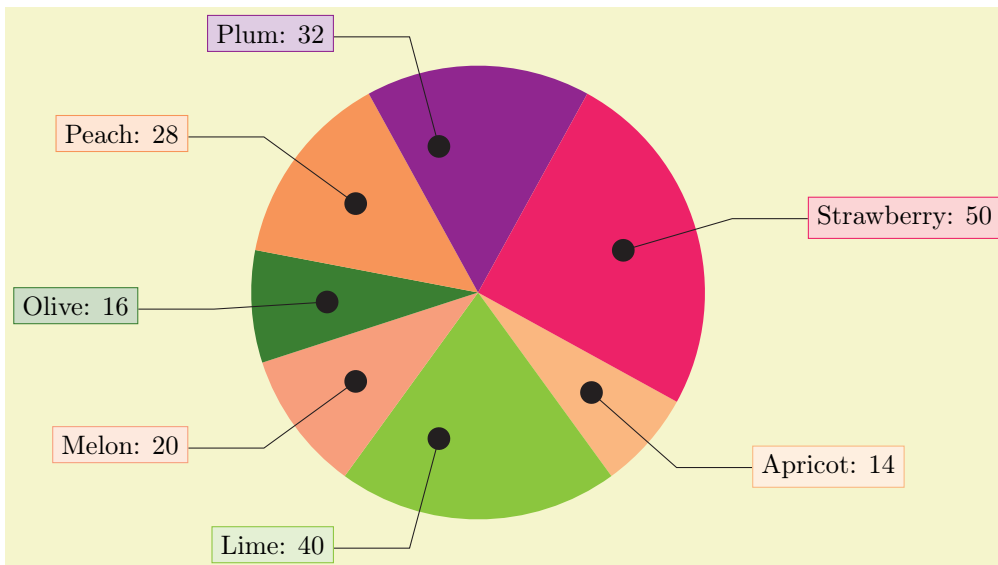
```

\usepackage {etoolbox} \usetikzlibrary {decorations.markings}
\begin{tikzpicture}
\def\WClegendcolumns{2}%specify the number of columns in the legend
\def\WClegendrow#1#2#3{\tikz\fill[#1] (0,0) circle[radius=0.15]; & #2 & $#3$}
\wheelchart[
  data={\WCperc},
  data style={outer xsep=4pt},
  legend entry={
    |csgdef{\WClegend\WCcount}{|}
    |csgappto{\WClegend\WCcount}{\WClegendrow}
    |cszappto{\WClegend\WCcount}{\WCvarB}{\WCvarC}{\WCvarA}}
  },
  legend={
    |def\WClegend{}
    |pgfmathsetmacro{\WClegendrows}{int(ceil(\WCtotalcount/\WClegendcolumns))}
    |foreach\n in {1,...,\WClegendrows}{
      |foreach\k in {1,...,\WClegendcolumns}{
        |pgfmathparse{int(\n+(\k-1)*\WClegendrows)}
        |ifnum\pgfmathresult>\WCtotalcount
          |gappto\WClegend{ & & }
        |else
          |gappto\WClegend{\csname \WClegend}
          |zappto\WClegend{\pgfmathresult}
          |gappto\WClegend{\endcsname}
        |fi
        |ifnum\k=\WClegendcolumns
          |gappto\WClegend{||}
        |else
          |gappto\WClegend{ & }
        |fi
      }
    }
    |node[anchor=north,draw,rounded corners,thick] at (0,-4.5) {%
      |begin{tabular}{*{\WClegendcolumns}{l\theta{ }lr}}%
      |WClegend%
      |end{tabular}%
    };
  },
  lines=0.5,
  lines ext=1,
  lines ext bottom dir=left,
  lines ext dirsep=1,
  lines ext fixed,
  lines ext top dir=right,
  lines sep=0,
  lines style={
    \WCvarB,
    postaction=decorate,
    decoration={
      markings,
      mark=at position 1 with {
        |fill[\WCvarB] (0,0) circle[radius=0.15];
      }
    }
  },
  start angle=331.2
]{\exampleforthismanual}
\wheelchart[
  data={},
  radius={1.5}{2},
  slices style=\WCvarB!70,
  start angle=331.2
]{\exampleforthismanual}
\end{tikzpicture}

```



```
\begin{tikzpicture}
\wheelchart[
  data style={
    inner sep=0pt,
    shift={(0,0.1)}
  },
  lines,
  lines ext=1.2,
  lines ext bottom dir=right,
  lines ext dirsep=1,
  %lines ext fixed,
  lines ext left anchor={base west},
  lines ext right anchor={base east},
  lines ext top dir=left,
  lines sep=-0.5,
  %lines style=\WCvarB,
  start angle=331.2
]{\exampleforthismanual}
\end{tikzpicture}
```



```
\usetikzlibrary {decorations.markings}
\begin{tikzpicture}
\wheelchart[
  data={\WCvarC: \WCvarA},
  data angle shift=\WCvarG,
  data style={draw=\WCvarB,fill=\WCvarB!20},
  lines=1.5,
  lines ext=1,
  lines sep=-1,
  lines style={
    Black,
    postaction=decorate,
    decoration={
      markings,
      mark=at position 0 with {
        \fill[Black] (0,0) circle[radius=0.15];
      }
    }
  },
  pie,
  start angle=331.2
]{\exampleforthismanual}
\end{tikzpicture}
```

`/wheelchart/lines sep={\value}`

(no default, initially 0.2)

This key defines the distance between the wheelchart and the start of the lines.

`/wheelchart/lines style={\langle options \rangle}` (style, no default, initially empty)

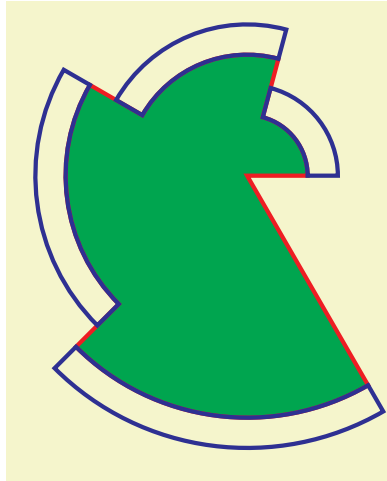
This key accepts a list of keys which will be applied to the lines drawn by the key `lines`.

`/wheelchart/middle={\langle text \rangle}` (no default)

This key contains the $\langle text \rangle$ which will be placed at the center of the wheelchart. The $\langle text \rangle$ is placed in a node. The style of this node is given as follows. First, the option `align=center` is given. Thereafter, the style of the key `middle style` is added.

`/wheelchart/middle fill={\langle options \rangle}` (style, no default, initially empty)

If this key is set then the middle of the wheelchart will be filled with this style.



```
\begin{tikzpicture}
\wheelchart[
  counterclockwise,
  data={},
  middle fill={
    Green,
    draw=Red,
    ultra thick
  },
  radius={0.8*\WCcount}
  {0.4+0.8*\WCcount},
  slices style={
    draw=Blue,
    fill=none,
    ultra thick
  },
  start angle=0,
  total angle=300,
  total count=4,
  value=1
]{}
\end{tikzpicture}
```

`/wheelchart/middle style={\langle options \rangle}` (style, no default, initially empty)

This key accepts a list of keys which will be applied to the node where the contents of the key `middle` is placed.

`/wheelchart/name={\langle name \rangle}` (no default, initially `wheelchart@name`)

This key defines the $\langle name \rangle$ of the local bounding box around the wheelchart.

`/wheelchart/outer radius={\langle value \rangle}` (no default, initially 3)

The $\langle value \rangle$ of this key defines the outer radius of the wheelchart.

`/wheelchart/perc precision={\langle number \rangle}` (no default, initially 0)

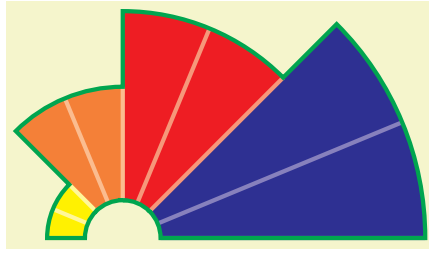
This key defines the number of decimals up to which the percentage in the macros `\WCperc` and `\WCpercentagerounded` are rounded.

`/wheelchart/pie={\langle boolean \rangle}` (default true, initially false)

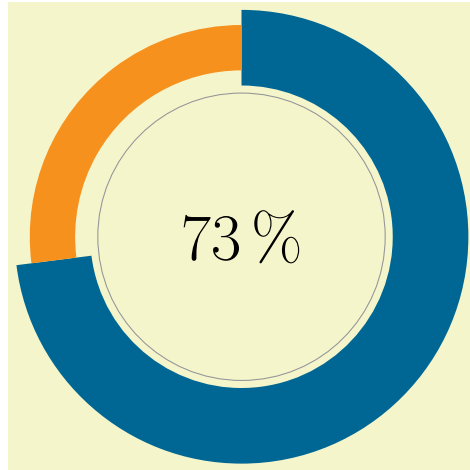
If true, the inner radius of the wheelchart is set to 0.

`/wheelchart/radius={\langle inner radius \rangle}{\langle outer radius \rangle}` (no default)

This key defines the inner and outer radius of the wheelchart.



```
\begin{tikzpicture}
\wheelchart[
  contour={Green,ultra thick},
  data={},
  radius={0.5}{\WCcount},
  slices style=\WCvarA,
  start angle=180,
  total angle=180,
  value=2,
  wheel lines={\WCvarA!50,ultra thick}
]{Yellow,Orange,Red,Blue}
\end{tikzpicture}
```

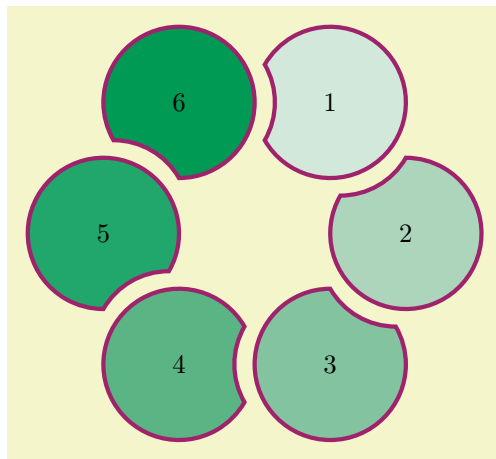


```
\usepackage{siunitx}
\begin{tikzpicture}
\def\n{73}
\wheelchart[
  data={},
  middle={{\Huge\qty{\n}{\percent}}},
  radius={2.5-\WCvarC}{2.5+\WCvarC}
]{%
  \n/MidnightBlue/0.5,
  {100-\n}/BurntOrange/0.3%
}
\draw[Gray](0,0)circle[radius=1.9];
\end{tikzpicture}
```

`/wheelchart/slices={\langle path \rangle}`

(no default)

If this key is set then the shape of the slices of the wheelchart is defined by $\langle path \rangle$.



```
\begin{tikzpicture}
\wheelchart[
  data={},
  radius={2}{2},
  slices={(90:1) arc[start angle=-30,
  end angle=-90,radius=1]
  arc[start angle=-210,
  end angle=90,radius=1]--cycle;},
  slices style={
    /utils/exec={
      \pgfmathsetmacro
        {\WCcolornumber}
        {(\WCcount/
          \WCtotalcount)*100}
    },
    ForestGreen!\WCcolornumber,
    draw=RedViolet,
    ultra thick
  },
  total count=6,
  value=1,
  wheel data=\WCcount
]{}
\end{tikzpicture}
```




```

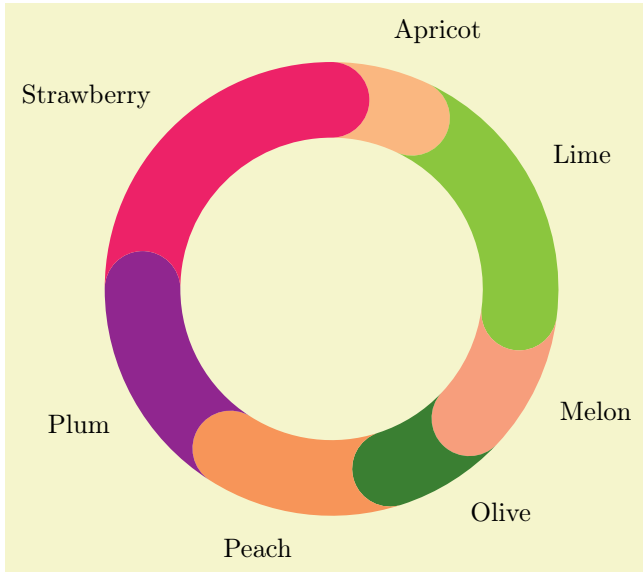
\begin{tikzpicture}
\wheelchart[
  radius={2.7}{3.1},
  slices={(0,-0.3)--(0.3,0)--(0,0.3)
  --cycle;},
  value=1
]{\exampleforthismanual}
\wheelchart[
  data={},
  value=1
]{\exampleforthismanual}
\wheelchart[
  data={},
  radius={2}{2},
  slices={(0,0) circle[radius=0.4];},
  slices style=White,
  value=1
]{\exampleforthismanual}
\wheelchart[
  data={},
  radius={2}{2},
  slices={(0,0) circle[radius=0.3];},
  value=1,
  wheel data=\WCcount
]{\exampleforthismanual}
\end{tikzpicture}

```

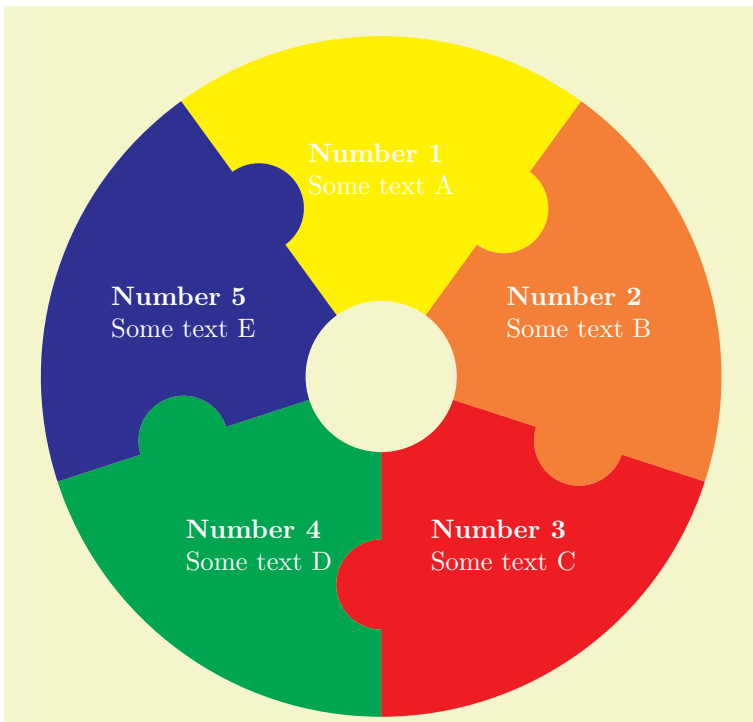
`/wheelchart/slices arc={⟨value 1⟩}{⟨value 2⟩}` (no default)

This key sets both `slices end arc` and `slices start arc`. The effect of `⟨value 1⟩` and `⟨value 2⟩` is shown in the table below.

	<code>⟨value 2⟩ = -0.5</code>	<code>⟨value 2⟩ = 0</code>	<code>⟨value 2⟩ = 0.5</code>
<code>⟨value 1⟩ = 2</code>			
<code>⟨value 1⟩ = 1</code> <code>⟨value 1⟩ = -1</code>			
<code>⟨value 1⟩ = -2</code>			



```
\begin{tikzpicture}
\wheelchart[
  slices arc={1}{0}
]{\exampleforthismanual}
\end{tikzpicture}
```

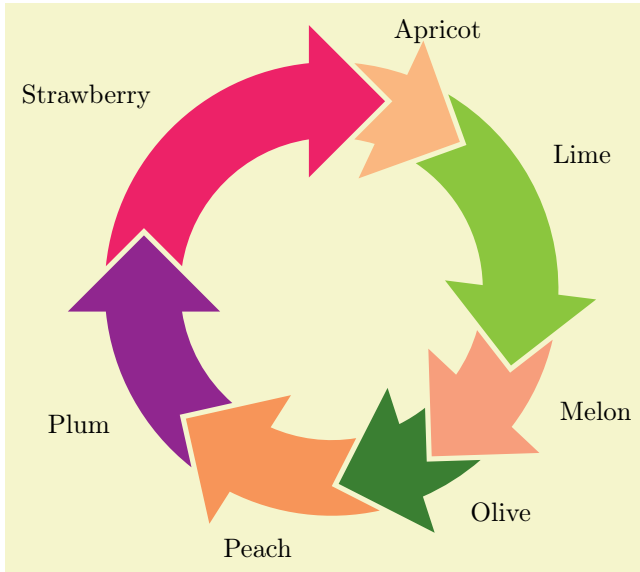


```
\begin{tikzpicture}
\wheelchart[
  data={},
  radius={1}{4.5},
  slices arc={1}{0.66},
  slices style=\WCvarA,
  start half,
  value=1,
  wheel data={%
    \textbf{Number \WCcount}}\|%
    \WCvarB%
  },
  wheel data pos=0.5,
  wheel data style=White
]{%
  Yellow/{Some text A},
  Orange/{Some text B},
  Red/{Some text C},
  Green/{Some text D},
  Blue/{Some text E}%
}
\end{tikzpicture}
```

`/wheelchart/slices arrow={\value 1}}{\value 2}}`

(no default)

This key is similar to the key `slices arc`.



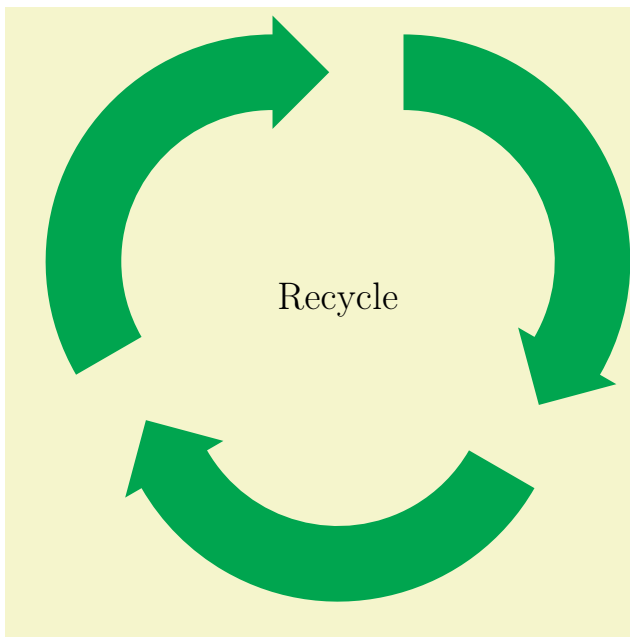
```
\begin{tikzpicture}
\wheelchart[
  gap=0.3,
  slices arrow={1}{-1}
]{\exampleforthismanual}
\end{tikzpicture}
```

`/wheelchart/slices end arc={\value 1}}{\value 2}}` (no default)

This key is similar to the key `slices arc` but only sets the end of the slice.

`/wheelchart/slices end arrow={\value 1}}{\value 2}}` (no default)

This key is similar to the key `slices arrow` but only sets the end of the slice.



```
\begin{tikzpicture}
\wheelchart[
  data={},
  explode=1,
  middle={\Large Recycle},
  slices end arrow={1}{-0.5},
  slices style=Green,
  total count=3,
  value=1
]{\}
\end{tikzpicture}
```

`/wheelchart/slices start arc={\value 1}}{\value 2}}` (no default)

This key is similar to the key `slices arc` but only sets the start of the slice.

`/wheelchart/slices start arrow={\value 1}}{\value 2}}` (no default)

This key is similar to the key `slices arrow` but only sets the start of the slice.

`/wheelchart/slices style={\options}}` (style, no default, initially `\WCvarB`)

This key defines the style of the slices of the wheelchart.

`/wheelchart/start angle={\angle}}` (no default, initially 90)

This key defines the `\angle` in degrees at which the first slice of the wheelchart starts.

`/wheelchart/start half={⟨angle⟩}` (default 90)

If this key is set then the middle of the first slice of the wheelchart is positioned at $\langle angle \rangle$ in degrees.

`/wheelchart/title={⟨text⟩}` (no default)

This key contains the $\langle text \rangle$ which will be placed above the wheelchart. The $\langle text \rangle$ is placed in a node. The x coordinate of this node is the x coordinate of the center of the wheelchart, which is defined by the key `at`. In general, this is *not* the same as the x coordinate of the center of the local bounding box around the wheelchart. The y coordinate of this node is 0.5 above the north of the local bounding box around the wheelchart. The style of this node is given as follows. First, the options `anchor=south,align=center` are given. Thereafter, the style of the key `title style` is added.

`/wheelchart/title left={⟨text⟩}` (no default)

This key contains the $\langle text \rangle$ which will be placed above left of the wheelchart. The $\langle text \rangle$ is placed in a node. This node is placed 0.5 above the north west of the local bounding box around the wheelchart. The style of this node is given as follows. First, the options `anchor=south west,align=left` are given. Thereafter, the style of the key `title left style` is added.

`/wheelchart/title left style={⟨options⟩}` (style, no default, initially empty)

This key accepts a list of keys which will be applied to the node where the contents of the key `title left` is placed.

`/wheelchart/title style={⟨options⟩}` (style, no default, initially empty)

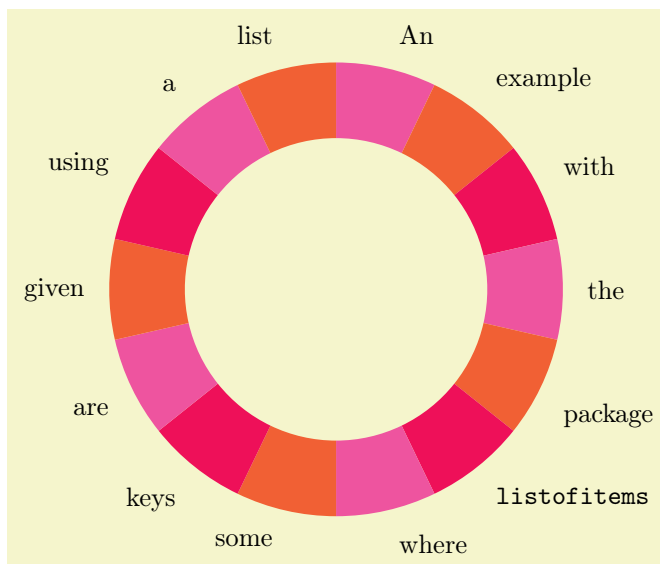
This key accepts a list of keys which will be applied to the node where the contents of the key `title` is placed.

`/wheelchart/total angle={⟨angle⟩}` (no default, initially 360)

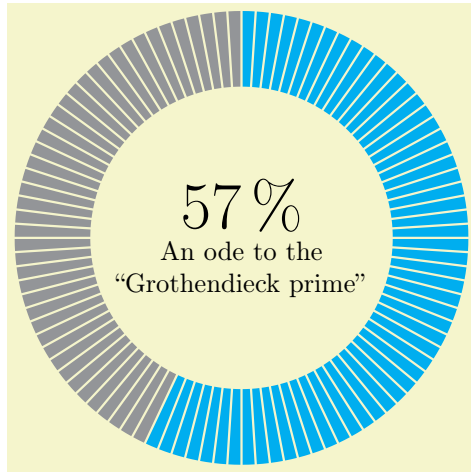
This key defines the total $\langle angle \rangle$ in degrees of the wheelchart.

`/wheelchart/total count={⟨number⟩}` (no default)

If this key is set then the number of slices of the wheelchart is determined by $\langle number \rangle$.



```
\usepackage {listofitems}
\readlist* \WCcolors{
  Rhodamine,RedOrange,OrangeRed
}
\setsepchar{ }
\readlist \WCdata{An example with the
  package \texttt{listofitems} where
  some keys are given using a list}
\begin{tikzpicture}
\wheelchart[
  data={\WCdata[\WCcount]},
  slices style={
    /utils/exec={
      \pgfmathsetmacro
        {\WCcolornumber}
        {\int (Mod({\WCcount-1},
          \WCcolorslen)+1)}
    },
    \WCcolors[\WCcolornumber]
  },
  total count={\WCdatalen},
  value=1
]{}
\end{tikzpicture}
```



```

\usepackage {siunitx}
\begin{tikzpicture}
\def\n{57}
\wheelchart[
  data={},
  gap=0.015,
  middle={%
    {\Huge\qty{\n}{\percent}}\|%
    An ode to the\|%
    "Grothendieck prime"%
  },
  slices style={
    /utils/exec={
      \ifnum\WCcount>\n
        \def\WCcolor{Gray}
      \else
        \def\WCcolor{Cyan}
      \fi
    },
    \WCcolor
  },
  total count=100,
  value=1
]{}
\end{tikzpicture}

```

`/wheelchart/value={\langle value \rangle}` (no default, initially `\WCvarA`)

This key defines the `\langle value \rangle` which corresponds to the size of each slice of the wheelchart.

`/wheelchart/wheel data={\langle text \rangle}` (no default)

This key contains the `\langle text \rangle` which will be placed on top of each slice of the wheelchart. The `\langle text \rangle` is placed in a node. The style of this node is given as follows. First, the option `align=left` is given. Thereafter, the style of the key `wheel data style` is added.

`/wheelchart/wheel data pos={\langle value \rangle}` (no default, initially 0.66)

The radius of the polar coordinate at which the contents of the key `wheel data` is placed is given by the convex combination `wheel data pos · outer radius + (1 - wheel data pos) · inner radius`.

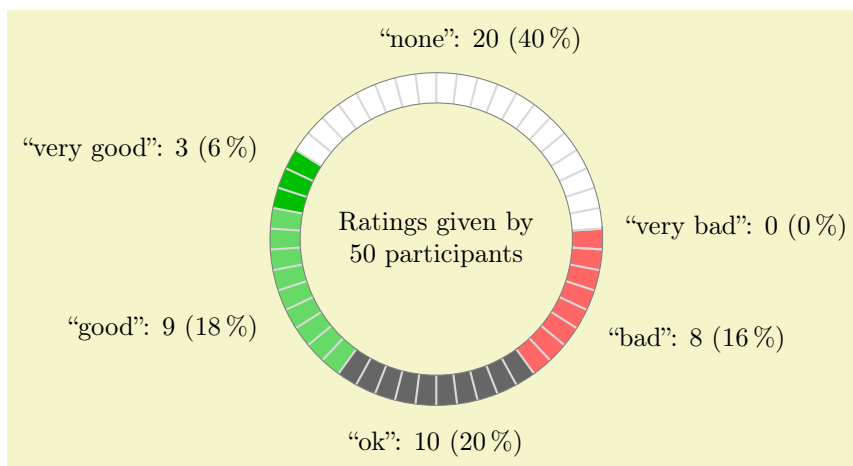
`/wheelchart/wheel data style={\langle options \rangle}` (style, no default, initially empty)

This key accepts a list of keys which will be applied to the node where the contents of the key `wheel data` is placed.

`/wheelchart/wheel lines={\langle options \rangle}` (style, no default, initially empty)

If this key is set then lines with the style determined by this key will be drawn inside the slices of the wheelchart. The number of these lines depends on the value of the key `value`.

Below is the example from [5, Subsection 7.6] recreated with the package `wheelchart`.



```

\begin{tikzpicture}
\colorlet{good}{green!75!black}
\colorlet{bad}{red}
\colorlet{neutral}{black!60}
\colorlet{none}{white}
\wheelchart[
  anchor xsep=15,
  contour=gray,
  data={"\WCvarC": \WCvarA} (\WCperc)},
  middle={Ratings given by \pgfmathprintnumber{\WCtotalnum}~participants},
  radius={1.8}{2.2},
  start half=270,
  wheel lines={black!15,thick}
]{}
  10/neutral/ok,
  9/good!60!white/good,
  3/good/{very good},
  20/none/none,
  0/bad/{very bad},
  8/bad!60!white/bad%
}
\end{tikzpicture}

```

References

- [1] Jake, *How can I produce a ‘ring (or wheel) chart’ like that on page 88 of the PGF manual?*, <https://tex.stackexchange.com/questions/17898/how-can-i-produce-a-ring-or-wheel-chart-like-that-on-page-88-of-the-pgf-manu/18105#18105>, 2011.
- [2] Jens-Uwe Morawski, *piechartMP*, Manual for Preliminary Version, <https://ctan.org/pkg/piechartmp>, 2002.
- [3] Dominique Rodriguez, Michael Sharpe, Herbert Voß, *pstricks-add additional Macros for pstricks*, Manual for version 3.92, <https://ctan.org/pkg/pstricks-add>, 2021.
- [4] Nicola L.C. Talbot, *User Manual for datatool bundle version 2.32*, <https://ctan.org/pkg/datatool>, 2019.
- [5] Till Tantau, *The TikZ and PGF Packages*, Manual for version 3.1.9a, <https://ctan.org/pkg/pgf>, 2021.
- [6] Yuan Xu, *Drawing Pie Chart by using pgf-pie*, Manual for version 0.6, <https://ctan.org/pkg/pgf-pie>, 2021.

Index

- anchor xsep key, 7
- anchor ysep key, 7
- at key, 11

- caption key, 11
- caption left key, 11
- caption left style key, 11
- caption style key, 11
- contour key, 12
- counterclockwise key, 12

- data key, 13
- data angle shift key, 13
- data sep key, 13
- data style key, 13

- `\exampleforthismanual`, 2
- expand list key, 13
- explode key, 15

- gap key, 16
- gap polar key, 16

- inner data key, 17
- inner data sep key, 18
- inner data style key, 18
- inner radius key, 18

- legend key, 18
- legend entry key, 18
- lines key, 18
- lines ext key, 19
- lines ext bottom dir key, 19
- lines ext dirsep key, 19
- lines ext fixed key, 19
- lines ext left anchor key, 19
- lines ext right anchor key, 19
- lines ext top dir key, 19
- lines sep key, 22
- lines style key, 23

- middle key, 23
- middle fill key, 23
- middle style key, 23

- name key, 23

- outer radius key, 23

- perc precision key, 23
- pie key, 23

- radius key, 23

- slices key, 24
- slices arc key, 25
- slices arrow key, 26
- slices end arc key, 27
- slices end arrow key, 27
- slices start arc key, 27
- slices start arrow key, 27
- slices style key, 27
- start angle key, 27
- start half key, 28

- title key, 28
- title left key, 28
- title left style key, 28
- title style key, 28
- total angle key, 28
- total count key, 28

- value key, 29

- `\WCcount`, 3
- `\WCdataangle`, 3
- `\WCmidangle`, 3
- `\WCperc`, 4
- `\WCpercentage`, 4
- `\WCpercentagerounded`, 5
- `\WCtotalcount`, 5
- `\WCtotalnum`, 5
- `\WCvarA`, 6
- `\WCvarB`, 6
- `\WCvarC`, 6
- `\WCvarD`, 6
- `\WCvarE`, 6
- `\WCvarF`, 6
- `\WCvarG`, 6
- `\WCvarH`, 6
- `\WCvarI`, 6
- `\WCvarJ`, 6
- `\WCvarK`, 6
- `\WCvarL`, 6
- `\WCvarM`, 6
- `\WCvarN`, 6
- `\WCvarO`, 6
- `\WCvarP`, 6
- `\WCvarQ`, 6
- `\WCvarR`, 6
- `\WCvarS`, 6
- `\WCvarT`, 6
- `\WCvarU`, 6
- `\WCvarV`, 6
- `\WCvarW`, 6
- `\WCvarX`, 6
- `\WCvarY`, 6
- `\WCvarZ`, 6
- wheel data key, 29
- wheel data pos key, 29
- wheel data style key, 29
- wheel lines key, 29
- `\wheelchart`, 2
- `/wheelchart/`
 - anchor xsep, 7

anchor ysep, 7
 at, 11
 caption, 11
 caption left, 11
 caption left style, 11
 caption style, 11
 contour, 12
 counterclockwise, 12
 data, 13
 data angle shift, 13
 data sep, 13
 data style, 13
 expand list, 13
 explode, 15
 gap, 16
 gap polar, 16
 inner data, 17
 inner data sep, 18
 inner data style, 18
 inner radius, 18
 legend, 18
 legend entry, 18
 lines, 18
 lines ext, 19
 lines ext bottom dir, 19
 lines ext dirsep, 19
 lines ext fixed, 19
 lines ext left anchor, 19
 lines ext right anchor, 19
 lines ext top dir, 19
 lines sep, 22
 lines style, 23
 middle, 23
 middle fill, 23
 middle style, 23
 name, 23
 outer radius, 23
 perc precision, 23
 pie, 23
 radius, 23
 slices, 24
 slices arc, 25
 slices arrow, 26
 slices end arc, 27
 slices end arrow, 27
 slices start arc, 27
 slices start arrow, 27
 slices style, 27
 start angle, 27
 start half, 28
 title, 28
 title left, 28
 title left style, 28
 title style, 28
 total angle, 28
 total count, 28
 value, 29
 wheel data, 29
 wheel data pos, 29
 wheel data style, 29
 wheel lines, 29