Quick Start

> Note: If you're viewing this notebook on `nbviewer.jupyter.org` some of the SVGs render improperly, even across cell output contexts. The bug is not in `itikz`.

Installation

Install TeX and pdf2svg

This is platform-dependent.

See:

- Texlive (https://www.tug.org/texlive/)
- pdf2svg (http://www.cityinthesky.co.uk/opensource/pdf2svg/)

Install itikz

```
pip install itikz
```

Usage

Load itikz. It's a jupter extension.

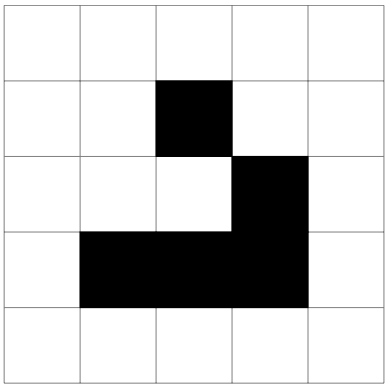
In [1]:
```
%load_ext itikz
```

Create a simple standalone document.

In [2]:

```
%%itikz
\documentclass[tikz]{standalone}
\begin{document}
\begin{tikzpicture}
\draw[help lines] grid (5, 5);
\draw[fill=black] (1, 1) rectangle (2, 2);
\draw[fill=black] (2, 1) rectangle (3, 2);
\draw[fill=black] (3, 1) rectangle (4, 2);
\draw[fill=black] (3, 2) rectangle (4, 3);
\draw[fill=black] (2, 3) rectangle (3, 4);
\end{tikzpicture}
\end{document}
```

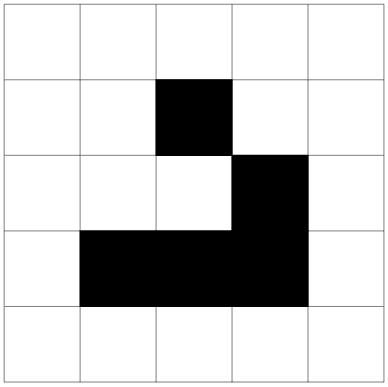
Out[2]:



In [3]:

```
%%itikz
\documentclass[tikz]{standalone}
\begin{document}
\begin{tikzpicture}
\draw[help lines] grid (5, 5);
\draw[fill=black] (1, 1) rectangle (2, 2);
\draw[fill=black] (2, 1) rectangle (3, 2);
\draw[fill=black] (3, 1) rectangle (4, 2);
\draw[fill=black] (3, 2) rectangle (4, 3);
\draw[fill=black] (2, 3) rectangle (3, 4);
\end{tikzpicture}
\end{document}
```

Out[3]:



The extension:

- Writes the cell as a `.tex` file;
- Runs `pdflatex` on the source;
- Runs `pdf2svg` on the generated pdf;
- Removes the intermediary artifacts.

By default, the filenames are the `md5` hash of the source. The extension uses the hash to see if regeneration is necessessary. If it's not, it just loads the SVG file.

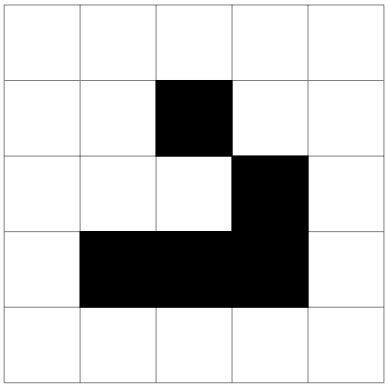
In [4]:
```
!ls *.svg *.tex
```

```
03f400523adb0f7d2fea15f4c4d6ad6e.svg  03f400523adb0f7d2fea15f4c4d6ad(
```

This is annoying sometimes if you want to look for a specific file outside of the notebook. So, you can prefix it, attaching semantical meaning.

In [5]:
```
%%itikz --file-prefix conway-

\documentclass[tikz]{standalone}
\begin{document}
\begin{tikzpicture}
\draw[help lines] grid (5, 5);
\draw[fill=black] (1, 1) rectangle (2, 2);
\draw[fill=black] (2, 1) rectangle (3, 2);
\draw[fill=black] (3, 1) rectangle (4, 2);
\draw[fill=black] (3, 2) rectangle (4, 3);
\draw[fill=black] (2, 3) rectangle (3, 4);
\end{tikzpicture}
\end{document}
```

Out[5]:



In [6]:
```
!ls *.svg *.tex
```

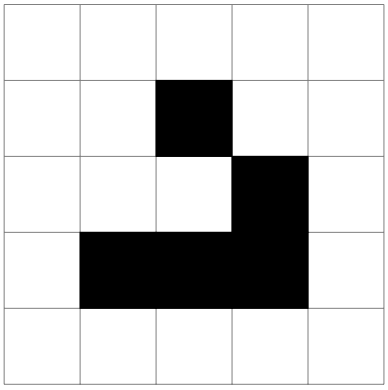
```
03f400523adb0f7d2fea15f4c4d6ad6e.svg
03f400523adb0f7d2fea15f4c4d6ad6e.tex
conway-04c3ec160176559dffa49ef2ac7746f2.svg
conway-04c3ec160176559dffa49ef2ac7746f2.tex
```

Of course, writing TikZ files entails lots of tiny tweaks, resulting in a lot of accumulated cruft. For development, you probably want to use your system temp directory to keep your project directory clean.

In [7]:
```
!rm -f *.svg *.tex
```

In [8]:
```
%%itikz --temp-dir --file-prefix conway-

\documentclass[tikz]{standalone}
\begin{document}
\begin{tikzpicture}
\draw[help lines] grid (5, 5);
\draw[fill=black] (1, 1) rectangle (2, 2);
\draw[fill=black] (2, 1) rectangle (3, 2);
\draw[fill=black] (3, 1) rectangle (4, 2);
\draw[fill=black] (3, 2) rectangle (4, 3);
\draw[fill=black] (2, 3) rectangle (3, 4);
\end{tikzpicture}
\end{document}
```

Out[8]:



In [9]:
```
!ls *.svg *.tex
```

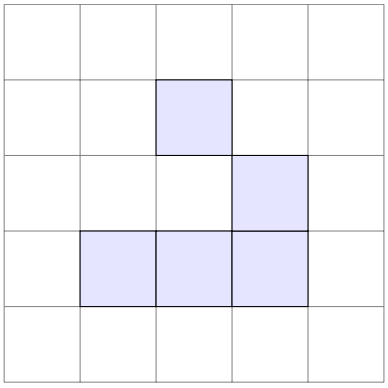
```
ls: cannot access '*.svg': No such file or directory
ls: cannot access '*.tex': No such file or directory
```

To make it easier to switch from development to production mode, setting the `ITIKZ_TEMP_DIR` environmental to any value enables `--temp-dir`.

In [10]:
```
import os
os.environ['ITIKZ_TEMP_DIR'] = '1'
```

In [11]:
```
%%itikz --file-prefix conway-
\documentclass[tikz]{standalone}
\begin{document}
\begin{tikzpicture}
\draw[help lines] grid (5, 5);
\draw[fill=blue!10] (1, 1) rectangle (2, 2);
\draw[fill=blue!10] (2, 1) rectangle (3, 2);
\draw[fill=blue!10] (3, 1) rectangle (4, 2);
\draw[fill=blue!10] (3, 2) rectangle (4, 3);
\draw[fill=blue!10] (2, 3) rectangle (3, 4);
\end{tikzpicture}
\end{document}
```

Out[11]:



In [12]:
```
!ls *.svg *.tex
```

```
ls: cannot access '*.svg': No such file or directory
ls: cannot access '*.tex': No such file or directory
```

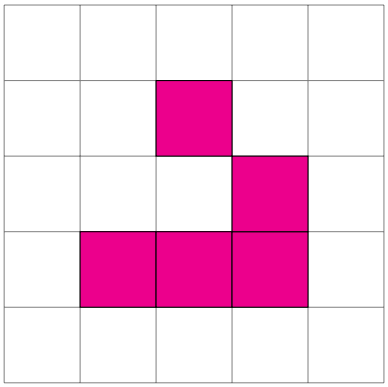
In [13]:
```
del os.environ['ITIKZ_TEMP_DIR']
```

Sometimes, you want to generate a TikZ document from a string, rather than a cell. You can do that using the line magic.

In [14]:
```
conway_str = r"""\documentclass[tikz]{standalone}
\begin{document}
\begin{tikzpicture}
\draw[help lines] grid (5, 5);
\draw[fill=magenta] (1, 1) rectangle (2, 2);
\draw[fill=magenta] (2, 1) rectangle (3, 2);
\draw[fill=magenta] (3, 1) rectangle (4, 2);
\draw[fill=magenta] (3, 2) rectangle (4, 3);
\draw[fill=magenta] (2, 3) rectangle (3, 4);
\end{tikzpicture}
\end{document}"""
```

In [15]:
```
%itikz --temp-dir --file-prefix conway- conway_str
```

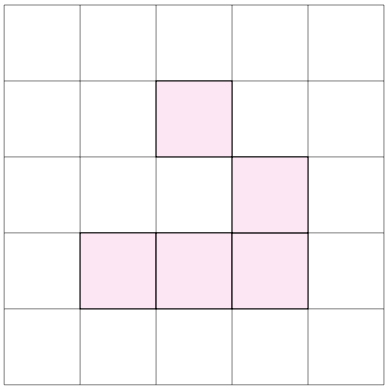
Out[15]:



Generally, string-generation is bad. One useful thing you can do without it is use an implicit `tikzpicture` environment.

In [16]:
```
%%itikz --file-prefix implicit-demo- --implicit-pic
\draw[help lines] grid (5, 5);
\draw[fill=magenta!10] (1, 1) rectangle (2, 2);
\draw[fill=magenta!10] (2, 1) rectangle (3, 2);
\draw[fill=magenta!10] (3, 1) rectangle (4, 2);
\draw[fill=magenta!10] (3, 2) rectangle (4, 3);
\draw[fill=magenta!10] (2, 3) rectangle (3, 4);
```

Out[16]:



Note that the resulting `tex` artifact is a full document so you can use it later when writing a `tex` document.

In [17]:
```
!cat implicit-demo-a6fdb3ecbc22048b7f090c20b5039b38.tex
```

```
cat: implicit-demo-a6fdb3ecbc22048b7f090c20b5039b38.tex: No such file
```

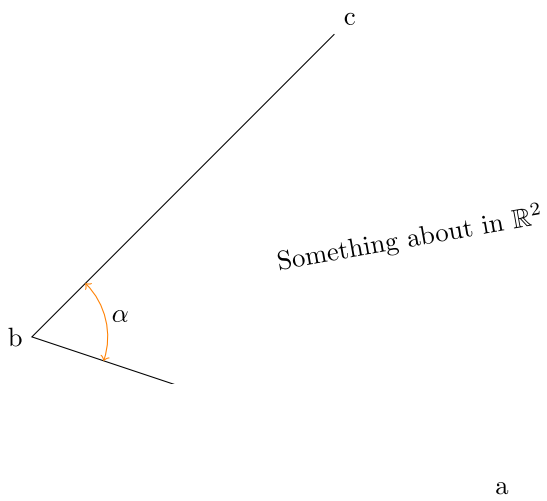
In [18]:
```
!rm implicit-demo*
```

In an `--implicit-pic` , it's often useful to:

- Set the `\tikzpicture[scale=X]` via `--scale=<X>` while iterating.
- Set the `\usepackage{X,Y,Z}` via `--tex-packages=<X,Y,Z>`
- Set the `\usetizlibrary{X,Y,Z}` via `--tiz-libraries=<X,Y,Z>`

In [19]:
```
%%itikz --temp-dir --implicit-pic --tikz-libraries=quotes,angles --tex-
% Example from Paul Gaborit
% http://www.texample.net/tikz/examples/angles-quotes/
\draw
    (3,-1) coordinate (a) node[right] {a}
    -- (0,0) coordinate (b) node[left] {b}
    -- (2,2) coordinate (c) node[above right] {c}
    pic["$\alpha$", draw=orange, <->, angle eccentricity=1.2, angle rac
    {angle=a--b--c};

\node[rotate=10] (r) at (2.5, 0.65) {Something about in $\mathbb{R}^2$}
```

Out[19]:



Sometimes, tikz-based packages don't use `tikzpicture` environments. To save a few keystrokes, you may want to use an implicit standalone flag. *Note: You have to use `--tex-packages=tikz` in this environment if you need tikz itself!*

In [20]:
```
%%itikz --temp-dir --implicit-standalone --tex-packages=smartdiagram,ar
\smartdiagramset{uniform sequence color=true,
sequence item border color=black,
sequence item font size=\footnotesize,
sequence item text color=white
}
\smartdiagram[sequence diagram]{
    $\mathbb{N}$,
    $\mathbb{Z}$,
    $\mathbb{Q}$,
    $\mathbb{R}$,
    $\mathbb{I}$,
    $\mathbb{C}$
}
```

Out[20]:

To help ensure that your tikz pictures stay aligned with your data -- and, to reduce the need for properly knowing PGF -- you can use jinja2 templates (http://jinja.pocoo.org/docs/latest/templates/)! For example, lets say you had five noes in a DAG, A, B, C, D, F. You could figure out positioning and such in the notebook, where your brain lives.

In [21]:
```
node_names = "ABCDEF"
nodes = {s: int(365/len(node_names) * i) for i, s in enumerate(node_nar
n = len(nodes)
nodes
```

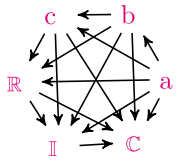
Out[21]:
```
{'A': 0, 'B': 60, 'C': 121, 'D': 182, 'E': 243, 'F': 304}
```

Then, you can interpret the cell magic source as a jinja2 template.

In [22]:

```
%%itikz --as-jinja --temp-dir
\documentclass[tikz]{standalone}
\usetikzlibrary{arrows,automata}
\definecolor{mymagenta}{RGB}{226,0,116}
\begin{document}
\begin{tikzpicture}[->,>=stealth',shorten >=1pt,auto,node distance=2.8c
                    semithick]
  \tikzstyle{every state}=[fill=mymagenta,draw=none,text=white]

  {% for name, angle in nodes.items() -%}
       \node[color=mymagenta] (v{{loop.index0}}) at ({{angle}}:1) {${{r
  {% endfor -%}

  {% for n1 in range(n) -%}
      {% for n2 in range(n) -%}
         {%if n1 < n2 -%}
             \path (v{{n1}}) edge (v{{n2}});
         {% endif -%}
      {% endfor -%}
  {% endfor -%}

\end{tikzpicture}

\end{document}
```

Out[22]:

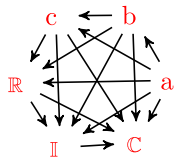


Which also works with the implicit environments.

In [23]:

```
%%itikz --as-jinja --temp-dir --tex-packages=tikz --tikz-libraries=arrc
\begin{tikzpicture}[->,>=stealth',shorten >=1pt,auto,node distance=2.8c
                    semithick]
  \tikzstyle{every state}=[fill=mymagenta,draw=none,text=white]

  {% for name, angle in nodes.items() -%}
      \node[color=red] (v{{loop.index0}}) at ({{angle}}:1) {${{name}}$
  {% endfor -%}

  {% for n1 in range(n) -%}
      {% for n2 in range(n) -%}
         {%if n1 < n2 -%}
             \path (v{{n1}}) edge (v{{n2}});
         {% endif -%}
      {% endfor -%}
  {% endfor -%}

\end{tikzpicture}
```

Out[23]:



Sometimes, you'll make mistakes. Debugging transpiled code is hard, especially without a mapping. To help, you can print the interpolated source.

In [24]:

```
%%itikz --as-jinja --print-jinja --temp-dir --as-jinja --tex-packages=t
\begin{tikzpicture}[->,>=stealth',shorten >=1pt,auto,node distance=2.8c
                    semithick]
  \tikzstyle{every state}=[fill=mymagenta,draw=none,text=white]

  {% for name, angle in nodes.items() -%}
       \node[color=red] (v{{loop.index0}}) at ({{angle}}:1) {${{name}}$
  {% endfor -%}

  {% for n1 in range(n) -%}
      {% for n2 in range(n) -%}
         {%if n1 < n2 -%}
             \path (v{{n1}}) edge (v{{n2}});
         {% endif -%}
      {% endfor -%}
  {% endfor %}

\end{tikzpicture}
```

```
\begin{tikzpicture}[->,>=stealth',shorten >=1pt,auto,node distance=2.8c
                    semithick]
  \tikzstyle{every state}=[fill=mymagenta,draw=none,text=white]

  \node[color=red] (v0) at (0:1) {$A$};
  \node[color=red] (v1) at (60:1) {$B$};
  \node[color=red] (v2) at (121:1) {$C$};
  \node[color=red] (v3) at (182:1) {$D$};
  \node[color=red] (v4) at (243:1) {$E$};
  \node[color=red] (v5) at (304:1) {$F$};
  \path (v0) edge (v1);
         \path (v0) edge (v2);
         \path (v0) edge (v3);
         \path (v0) edge (v4);
         \path (v0) edge (v5);
         \path (v1) edge (v2);
         \path (v1) edge (v3);
         \path (v1) edge (v4);
         \path (v1) edge (v5);
         \path (v2) edge (v3);
         \path (v2) edge (v4);
         \path (v2) edge (v5);
         \path (v3) edge (v4);
         \path (v3) edge (v5);
         \path (v4) edge (v5);


\end{tikzpicture}
```

Finally, its worth noting that jinja templating assumes a jinja2 file loader set in the `$CWD` . This means you can stick to the DRY principal with blocks and template extension.

In [25]:
```
%%writefile dag_demo.tex
\documentclass[tikz]{standalone}
\usetikzlibrary{arrows,automata}
\definecolor{mymagenta}{RGB}{226,0,116}
\begin{document}
\begin{tikzpicture}[->,>=stealth',shorten >=1pt,auto,node distance=2.8c
                    semithick]
  \tikzstyle{every state}=[fill=mymagenta,draw=none,text=white]

{% block content %}
{% endblock %}

\end{tikzpicture}
\end{document}
```

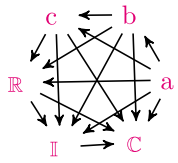
```
Writing dag_demo.tex
```

In [26]:
```
!ls dag_demo.tex
```

```
dag_demo.tex
```

In [27]:
```
%%itikz --as-jinja --temp-dir
{% extends "dag_demo.tex" %}
{% block content %}

{% for name, angle in nodes.items() %}
       \node[color=mymagenta] (v{{loop.index0}}) at ({{angle}}:1) {${{r
{% endfor -%}

{% for n1 in range(n) %}
    {% for n2 in range(n) %}
        {%if n1 < n2 %}
            \path (v{{n1}}) edge (v{{n2}});
        {% endif %}
    {% endfor -%}
{% endfor %}

{% endblock %}
```

Out[27]:

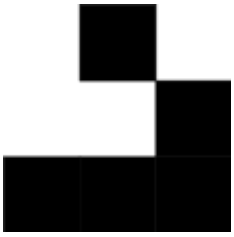


In [28]:
```
!rm dag_demo.tex # ignore this, it's just housekeeping
```

Sometimes, you may want to send a collaborator a quick snapshot of your work. But, lots of messaging windows won't accept SVG drag-and-drop. As a time saver, you can rasterize the SVG as a PNG (if you have cairosvg installed).

In [29]:
```
%%itikz --implicit-pic --temp-dir --rasterize
\draw[fill=black] (1, 1) rectangle (2, 2);
\draw[fill=black] (2, 1) rectangle (3, 2);
\draw[fill=black] (3, 1) rectangle (4, 2);
\draw[fill=black] (3, 2) rectangle (4, 3);
\draw[fill=black] (2, 3) rectangle (3, 4);
```

Out[29]:



Also, the latex command line error messages tend to be...verbose. By default, only the tail is shown.

In [30]:
```
%%itikz --implicit-pic
4$
```

```
l.6 \end{tikzpicture}

?
! Emergency stop.
<inserted text>
                $
l.6 \end{tikzpicture}

!  ==> Fatal error occurred, no output PDF file produced!
Transcript written on 26e1ca6384b15e1d929f6c01ce6ba32f.log.
```

But, if this isn't enough, you can see the whole message.

In [31]:
```
%%itikz --implicit-pic --full-error
4$
```

```
This is pdfTeX, Version 3.14159265-2.6-1.40.19 (TeX Live 2018/Debian)
 restricted \write18 enabled.
entering extended mode
(./26e1ca6384b15e1d929f6c01ce6ba32f.tex
LaTeX2e <2018-04-01> patch level 4
Babel <3.20> and hyphenation patterns for 84 language(s) loaded.
(/usr/share/texlive/texmf-dist/tex/latex/standalone/standalone.cls
Document Class: standalone 2018/03/26 v1.3a Class to compile TeX sub-fi
dalone
(/usr/share/texlive/texmf-dist/tex/latex/tools/shellesc.sty)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifluatex.sty)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifpdf.sty)
(/usr/share/texlive/texmf-dist/tex/generic/ifxetex/ifxetex.sty)
(/usr/share/texlive/texmf-dist/tex/latex/xkeyval/xkeyval.sty
(/usr/share/texlive/texmf-dist/tex/generic/xkeyval/xkeyval.tex
(/usr/share/texlive/texmf-dist/tex/generic/xkeyval/xkvutils.tex
(/usr/share/texlive/texmf-dist/tex/generic/xkeyval/keyval.tex))))
(/usr/share/texlive/texmf-dist/tex/latex/standalone/standalone.cfg)
(/usr/share/texlive/texmf-dist/tex/latex/base/article.cls
Document Class: article 2014/09/29 v1.4h Standard LaTeX document class
(/usr/share/texlive/texmf-dist/tex/latex/base/size10.clo))
(/usr/share/texlive/texmf-dist/tex/latex/pgf/frontendlayer/tikz.sty
(/usr/share/texlive/texmf-dist/tex/latex/pgf/basiclayer/pgf.sty
(/usr/share/texlive/texmf-dist/tex/latex/pgf/utilities/pgfrcs.sty
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfutil-commor
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfutil-commor
ex)) (/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfutil-l
(/usr/share/texlive/texmf-dist/tex/latex/ms/everyshi.sty))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfrcs.code.te
(/usr/share/texlive/texmf-dist/tex/latex/pgf/basiclayer/pgfcore.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/graphicx.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/graphics.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/trig.sty)
(/usr/share/texlive/texmf-dist/tex/latex/graphics-cfg/graphics.cfg)
(/usr/share/texlive/texmf-dist/tex/latex/graphics-def/pdftex.def)))
(/usr/share/texlive/texmf-dist/tex/latex/pgf/systemlayer/pgfsys.sty
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys.code
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfkeys.code.t
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfkeysfiltere
ex)) (/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgf.cfg
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys-pdfte
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys-commo
f)))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsyssoftpa
tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsysprotoc
tex)) (/usr/share/texlive/texmf-dist/tex/latex/xcolor/xcolor.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics-cfg/color.cfg))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcore.code
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmath.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathcalc.code.te
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathutil.code.te
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathparser.code
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.co
```

(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.ba
.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.tr
ric.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.ra
e.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.co
.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.ba
tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.ro
.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.mi
tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.in
thmetics.code.tex)))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfloat.code.t
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepoints
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepathco
code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepathus
.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorescopes
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoregraphi
ode.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoretransi
s.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorequick
)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreobject
ex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepathpr
.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorearrows
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreshade
)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreimage

(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreexterna
tex))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorelayers
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoretransp
ode.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepatter
tex)))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/modules/pgfmoduleshapes
) (/usr/share/texlive/texmf-dist/tex/generic/pgf/modules/pgfmoduleplot
)
(/usr/share/texlive/texmf-dist/tex/latex/pgf/compatibility/pgfcomp-vers
.sty)
(/usr/share/texlive/texmf-dist/tex/latex/pgf/compatibility/pgfcomp-vers
.sty)) (/usr/share/texlive/texmf-dist/tex/latex/pgf/utilities/pgffor.st
(/usr/share/texlive/texmf-dist/tex/latex/pgf/utilities/pgfkeys.sty

```
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfkeys.code.t
(/usr/share/texlive/texmf-dist/tex/latex/pgf/math/pgfmath.sty
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmath.code.tex))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgffor.code.te
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmath.code.tex))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/frontendlayer/tikz/tikz

(/usr/share/texlive/texmf-dist/tex/generic/pgf/libraries/pgflibraryplot
.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/modules/pgfmodulematrix
)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/frontendlayer/tikz/libra
zlibrarytopaths.code.tex))))
No file 26e1ca6384b15e1d929f6c01ce6ba32f.aux.
ABD: EveryShipout initializing macros
(/usr/share/texlive/texmf-dist/tex/context/base/mkii/supp-pdf.mkii
[Loading MPS to PDF converter (version 2006.09.02).]
) (/usr/share/texlive/texmf-dist/tex/latex/oberdiek/epstopdf-base.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/infwarerr.sty)
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/grfext.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/kvdefinekeys.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ltxcmds.sty)))
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/kvoptions.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/kvsetkeys.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/etexcmds.sty)))
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/pdftexcmds.sty)
(/usr/share/texlive/texmf-dist/tex/latex/latexconfig/epstopdf-sys.cfg)
! Missing $ inserted.
<inserted text>
                $
l.6 \end{tikzpicture}

?
! Emergency stop.
<inserted text>
                $
l.6 \end{tikzpicture}

!  ==> Fatal error occurred, no output PDF file produced!
Transcript written on 26e1ca6384b15e1d929f6c01ce6ba32f.log.
```

Finally, if you forget the usage, ask for help.

In [32]:
```
%itikz -h
```

```
usage: %%itikz [--temp-dir] [--file-prefix FILE_PREFIX] [--implicit-pic
               [--implicit-standalone] [--scale SCALE]
               [--tikz-libraries TIKZ_LIBRARIES] [--tex-packages TEX_PA
               [--as-jinja] [--print-jinja] [--rasterize] [--full-error
               [k]

Tikz to tex to SVG

positional arguments:
  k                     the variable in IPython with the string source

optional arguments:
  --temp-dir            emit artifacts to system temp dir
  --file-prefix FILE_PREFIX
                        emit artifacts with a path prefix
  --implicit-pic        wrap source in implicit tikzpicture document
  --implicit-standalone
                        wrap source in implicit document
  --scale SCALE         Set tikzpicture scale in --implicit-pic tmpl
  --tikz-libraries TIKZ_LIBRARIES
                        Comma separated list of tikz libraries to use
  --tex-packages TEX_PACKAGES
                        Comma separated list of tex packages to use
  --as-jinja            Interpret the source as a jinja2 template
  --print-jinja         Print interpolated jinja2 source then bail.
  --rasterize           Rasterize the svg with cairosvg
  --full-error          Emit the full error message
  -h, --help            show this help message
```