Ti*k*Z Tutorial

Nick Horelik

MIT

February 14, 2014

What Is TikZ?

- PGF: Portble Graphics Format (or "pretty, good, functional")
- Tikz: Tikz ist *kein* Zeichenprogramm
- Allows creation of vector graphic schemes, charts, diagrams, ...
- Placed inline in the middle of text
- In floats

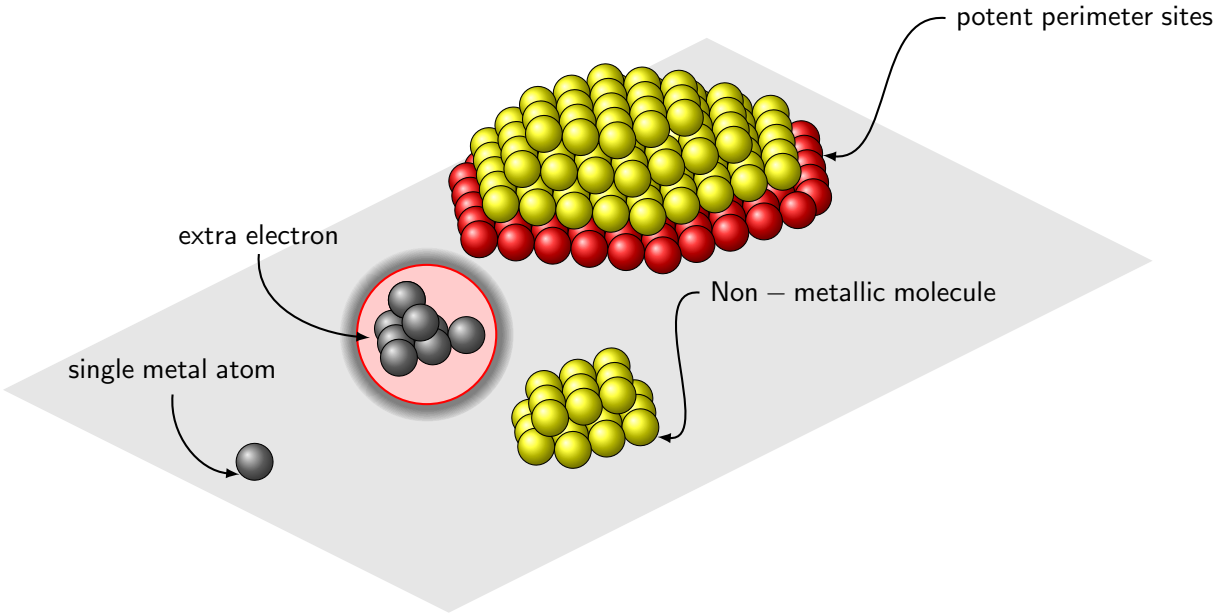


Figure: A tikzpicture

The Good and the Bad

- Pros
  - Programatically draw with exact precision
  - Consistent typography
  - Very fast for simple graphics
- Cons
  - Steep learning curve
  - No WYSIWYG
  - Changes require recompile

TONS of help online

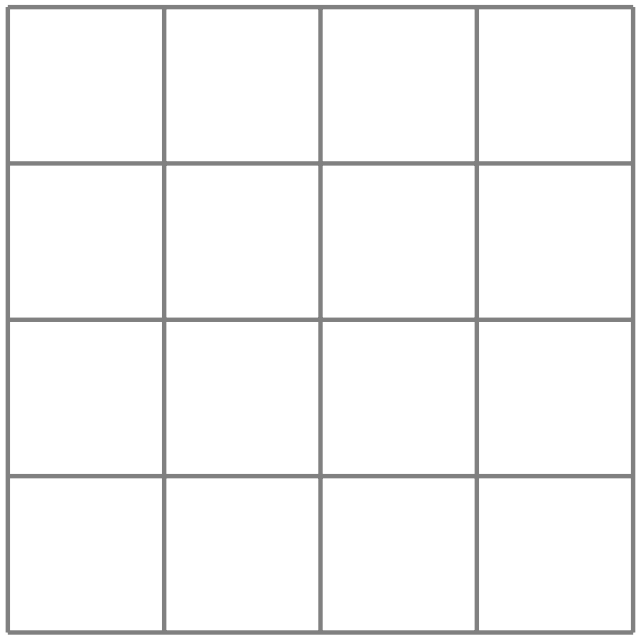
- Manual and primer: `www.ctan.org/pkg/pgf`
  - Search the document! *e.g.* search `/tikz/every node`
- Huge example repository: `www.texample.net/tikz/`
- Vibrant community for specific help: `tex.stackexchange.com/`

Google is your best friend!

(try searching for "tikz arrow head size")

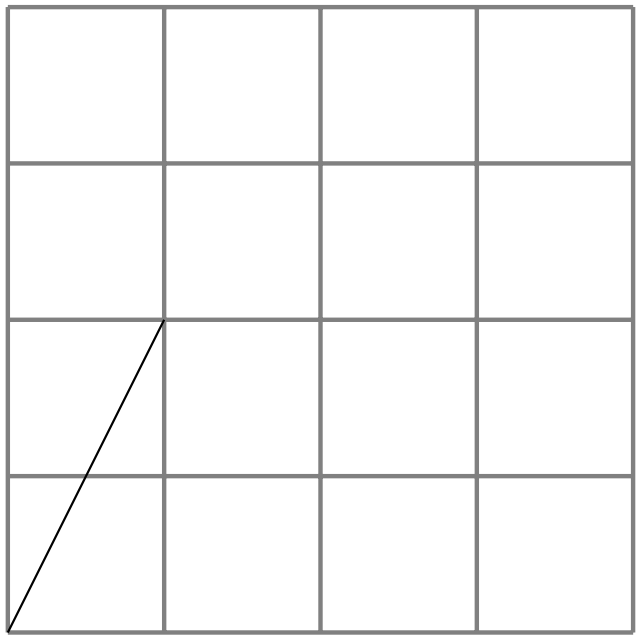
Example: Drawing a Path

```
\draw[help lines, thick] (0,0) grid (4,4);
```



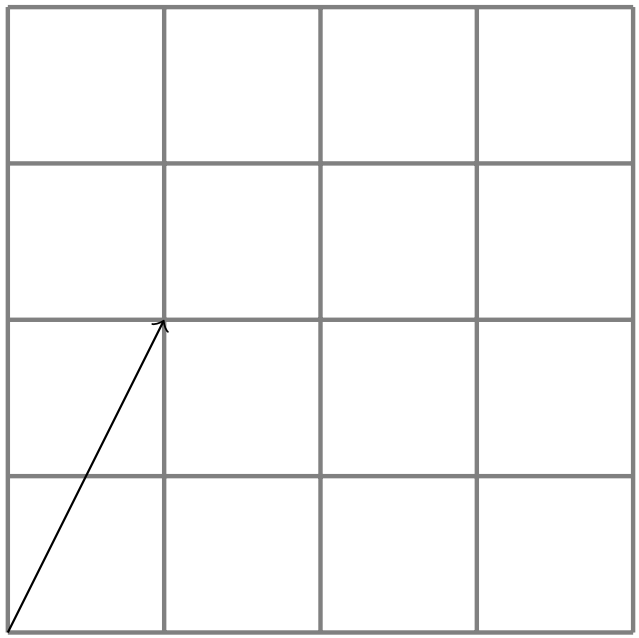
Example: Drawing a Path

```
\draw[help lines, thick] (0,0) grid (4,4);
\draw (0,0) -- (1,2);
```



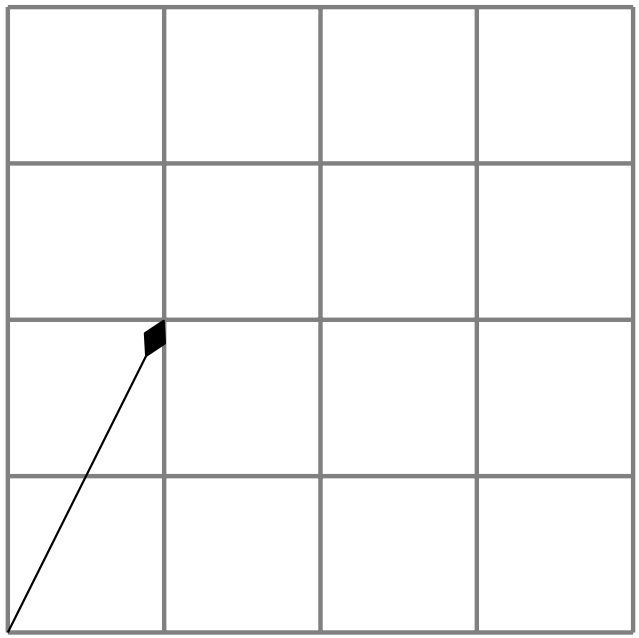
Example: Drawing a Path

```
\draw[help lines, thick] (0,0) grid (4,4);
\draw[->] (0,0) -- (1,2);
```



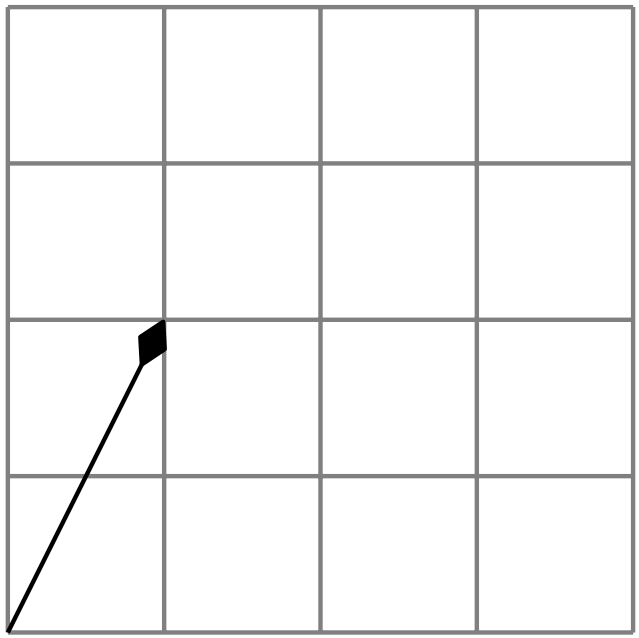
Example: Drawing a Path

```
\draw[help lines, thick] (0,0) grid (4,4);
\draw[->,>=diamond] (0,0) -- (1,2);
```



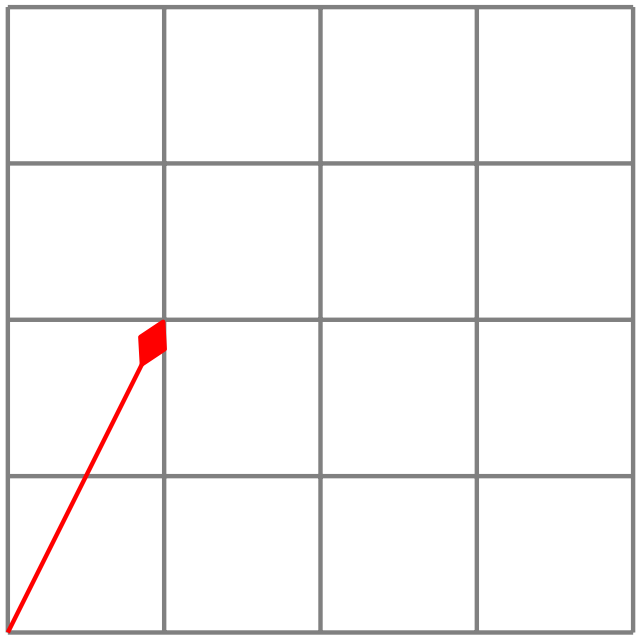
Example: Drawing a Path

```
\draw[help lines, thick] (0,0) grid (4,4);
\draw[->,>=diamond,thick] (0,0) -- (1,2);
```



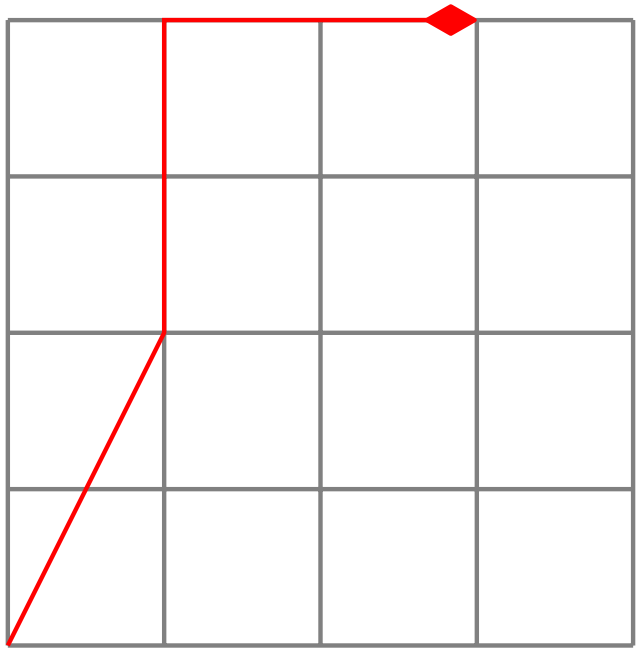
Example: Drawing a Path

```
\draw[help lines, thick] (0,0) grid (4,4);
\draw[->,>=diamond,thick,red] (0,0) -- (1,2);
```



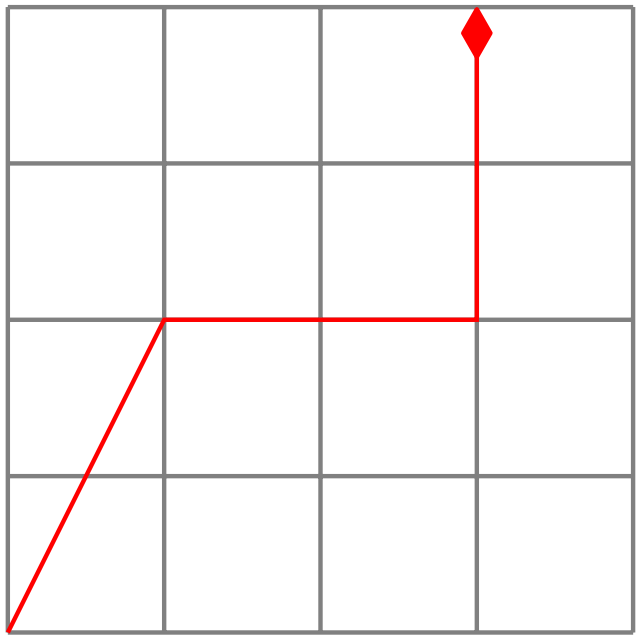
Example: Drawing a Path

```
\draw[help lines, thick] (0,0) grid (4,4);
\draw[->,>=diamond,thick,red] (0,0) -- (1,2) -| (3,4);
```



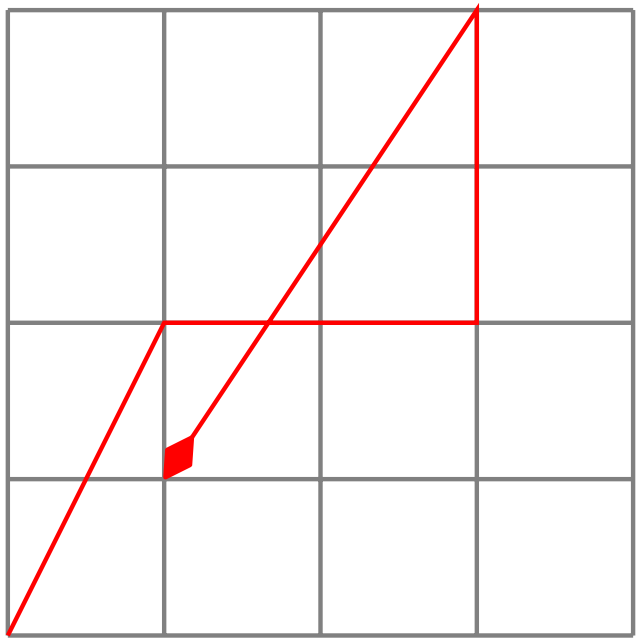
Example: Drawing a Path

```
\draw[help lines, thick] (0,0) grid (4,4);
\draw[->,>=diamond,thick,red] (0,0) -- (1,2) |- (3,4);
```



Example: Drawing a Path

```
\draw[help lines, thick] (0,0) grid (4,4);
\draw[->,>=diamond,thick,red] (0,0) -- (1,2) -| (3,4) -- (1,1);
```

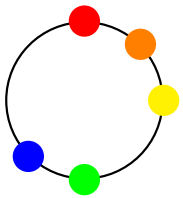


Tikz Core Concepts: Points

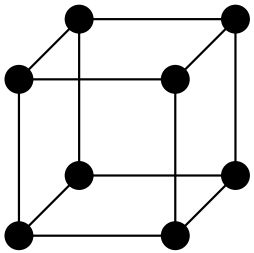
- Points: `(1cm,2pt)`
- Relative points:

```
(5,5)     +(1,0)    ++(1,1)     +(0,1)
  ↓         ↓         ↓           ↓
(5,5)      (6,0)     (6,6)       (6,7)
```

- Named locations: `node.south`

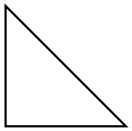


- 3D points: `(1,1,2)`



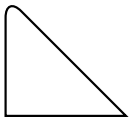
Tikz Core Concepts: Idea of Paths

- Series of straight or curved lines
- `\path(2em,0) -- (0,0) -- (0,2em) -- cycle`



- Can *act* on paths: `draw`, `fill`, `shade`, `clip`

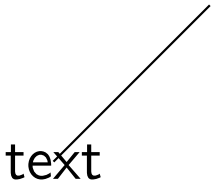
- Attributes can be applied mid-path



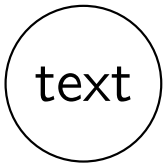
Tikz Core Concepts: Key-Value Parameters, Nodes

- Attributes are set everywhere with `key=value`

- Nodes are inserted at the current position of a path, or at a specified location



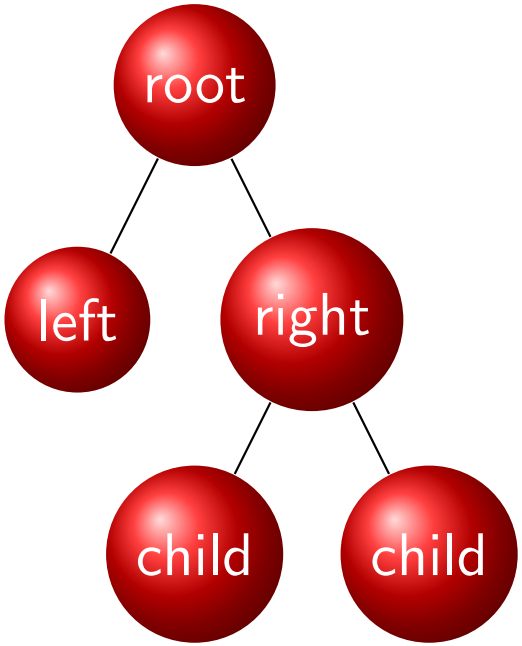
- Nodes can have options too



- Nodes can be named to reference their coordinates

Tikz Core Concepts: Trees

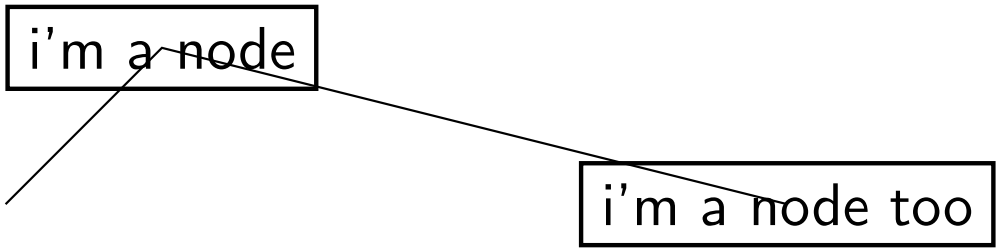
- Nodes can be arranged in trees automatically



Tikz Core Concepts: Scopes

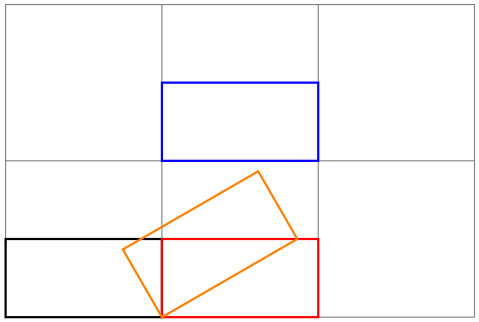
- Attributes can be applied to all objects in a scope

- Check out `/tikz/every path`, and `/tikz/every node`



Tikz Core Concepts: Transformations

- Coordinates can be transformed

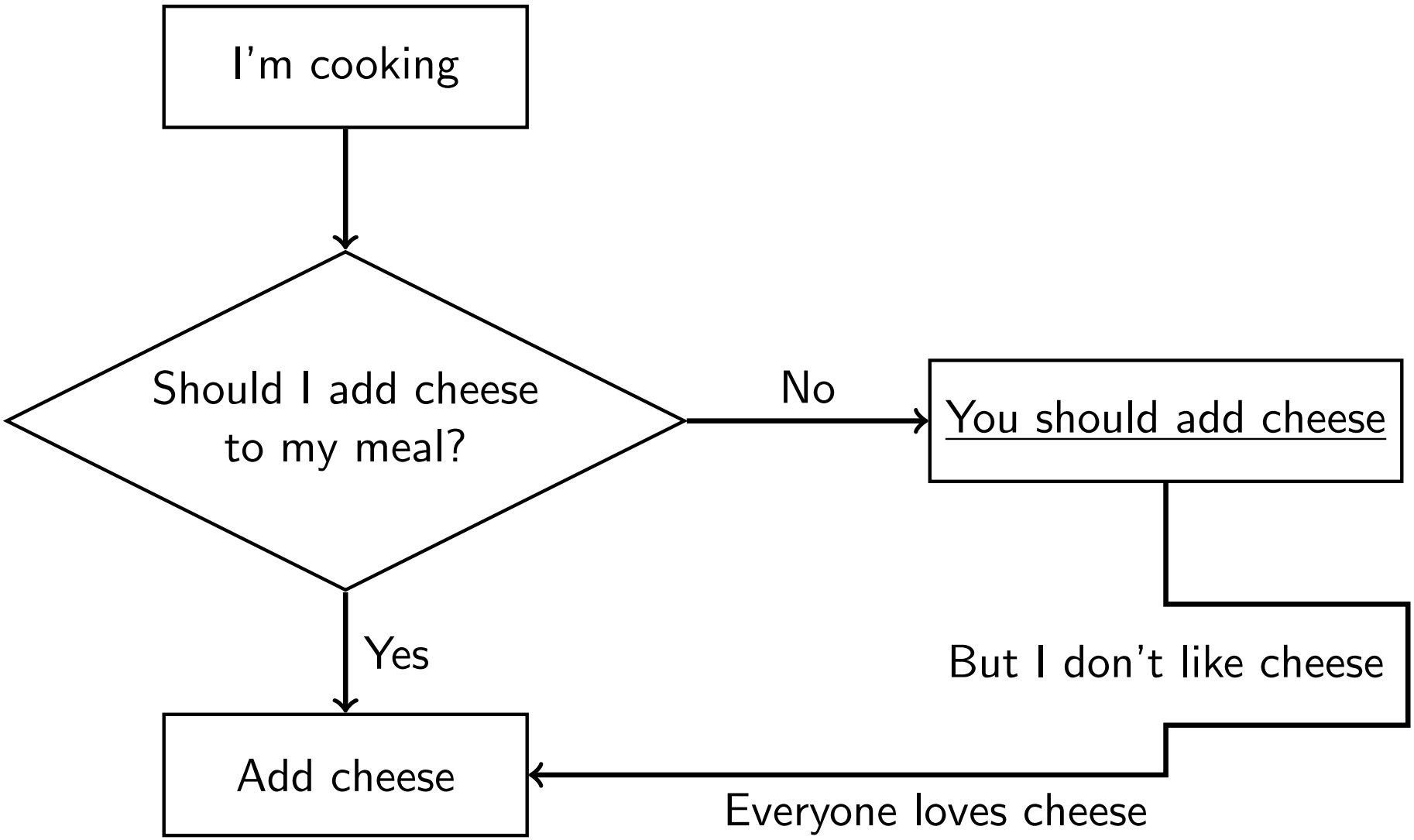


- Nodes can be moved too

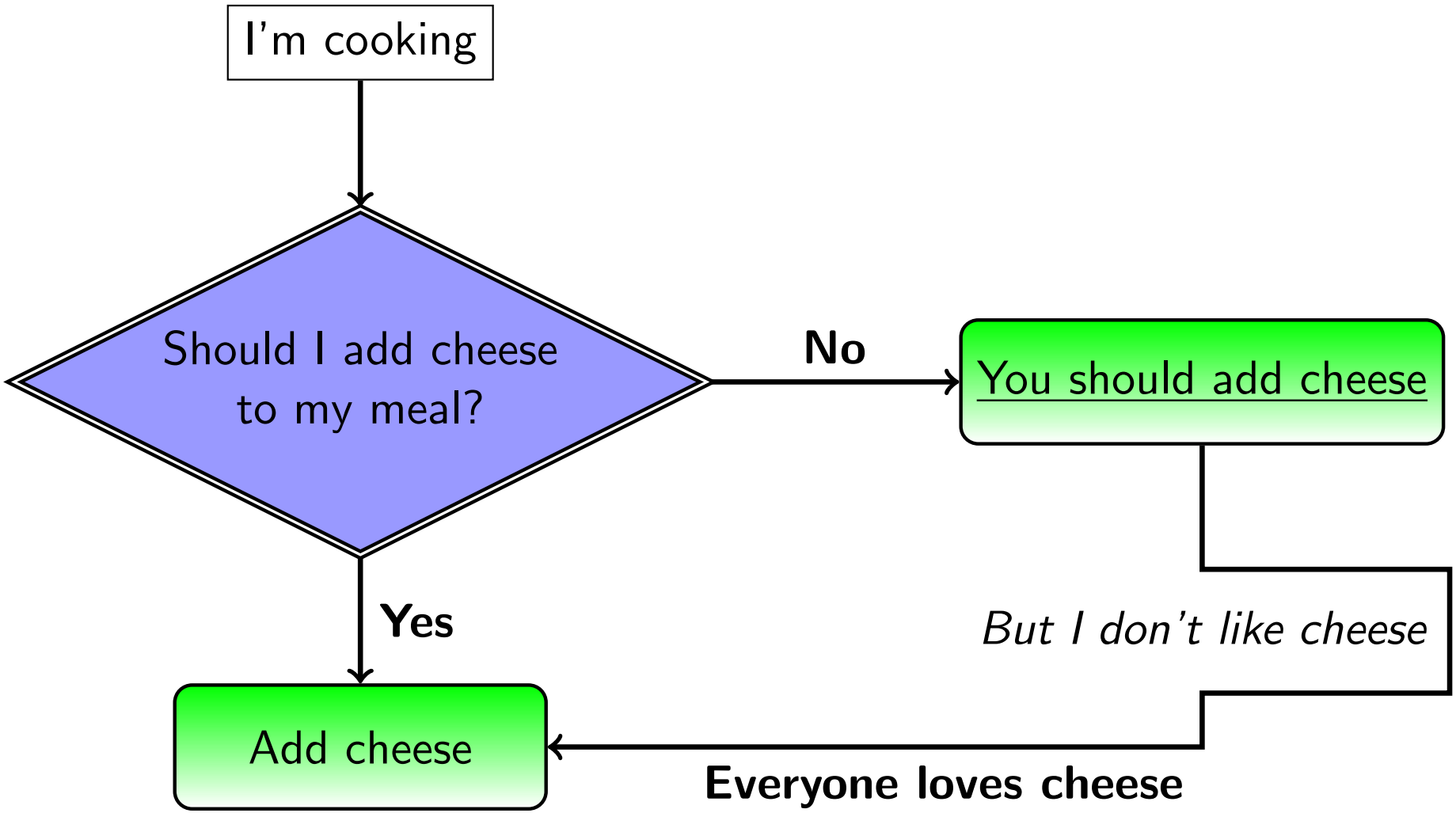
i'm a node

i'm a node too

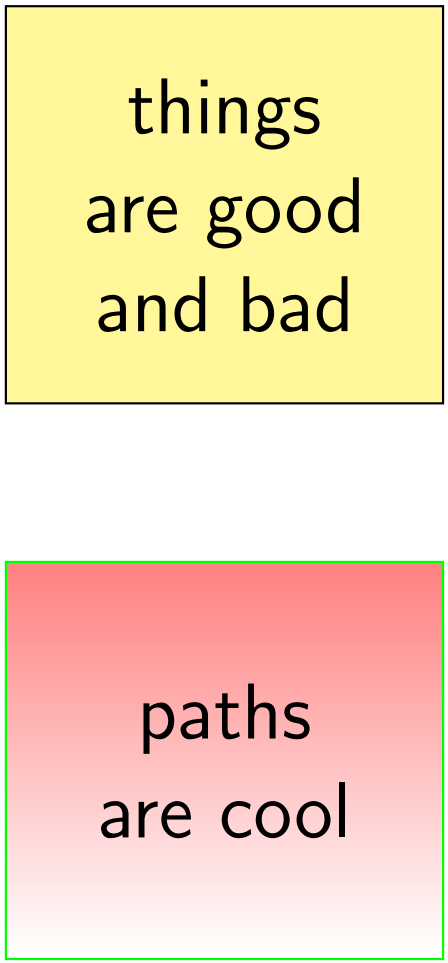
Example: Decision Tree



Styles, Libraries, Colors

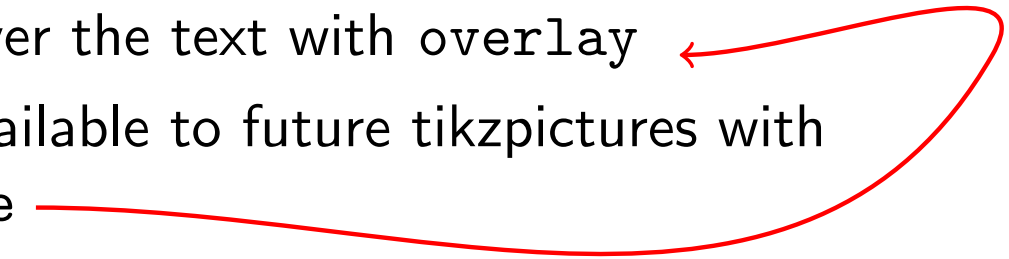


Example: Hyperlinks



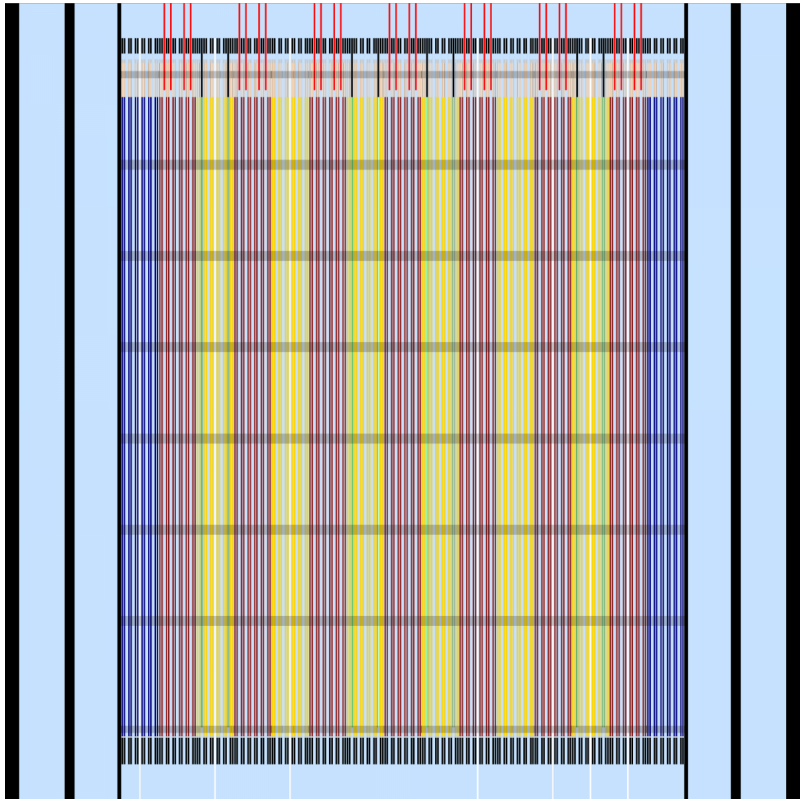
Text Overlays

- You can use tikz over the text with `overlay`
- Node names are available to future tikzpictures with `remember picture`



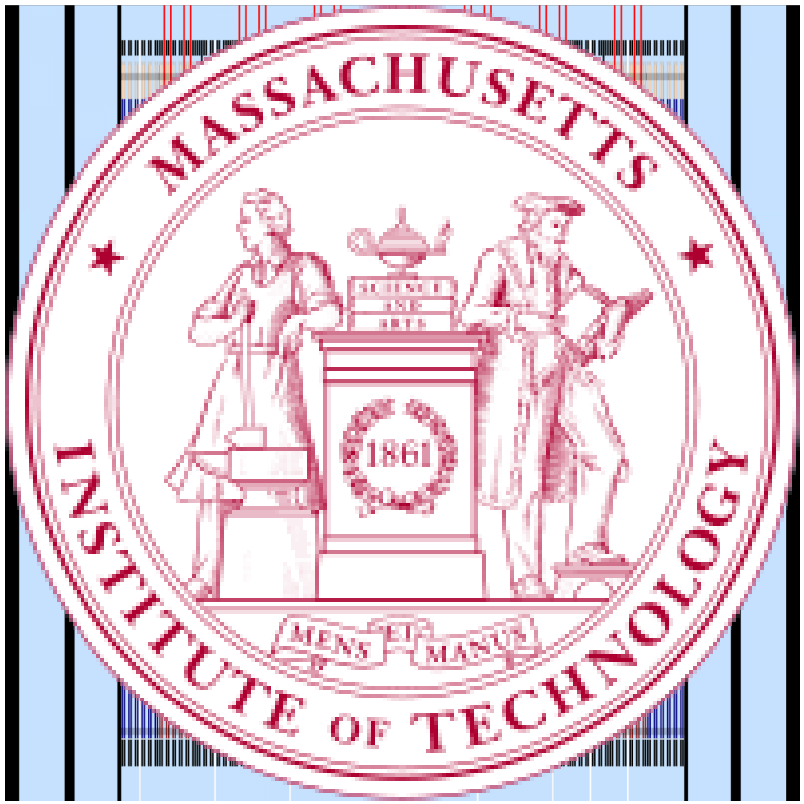
Stacked Figures in Beamer

- Stacking pictures in Beamer is easy with overlays

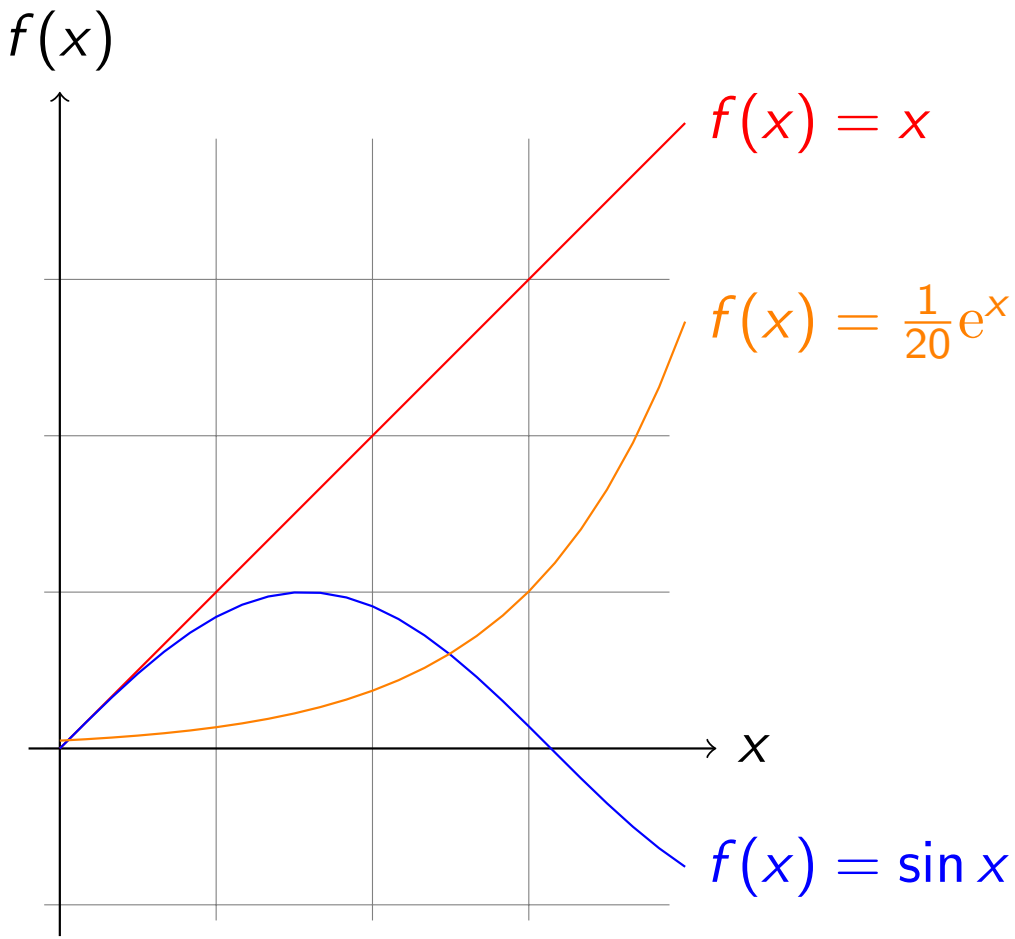


Stacked Figures in Beamer

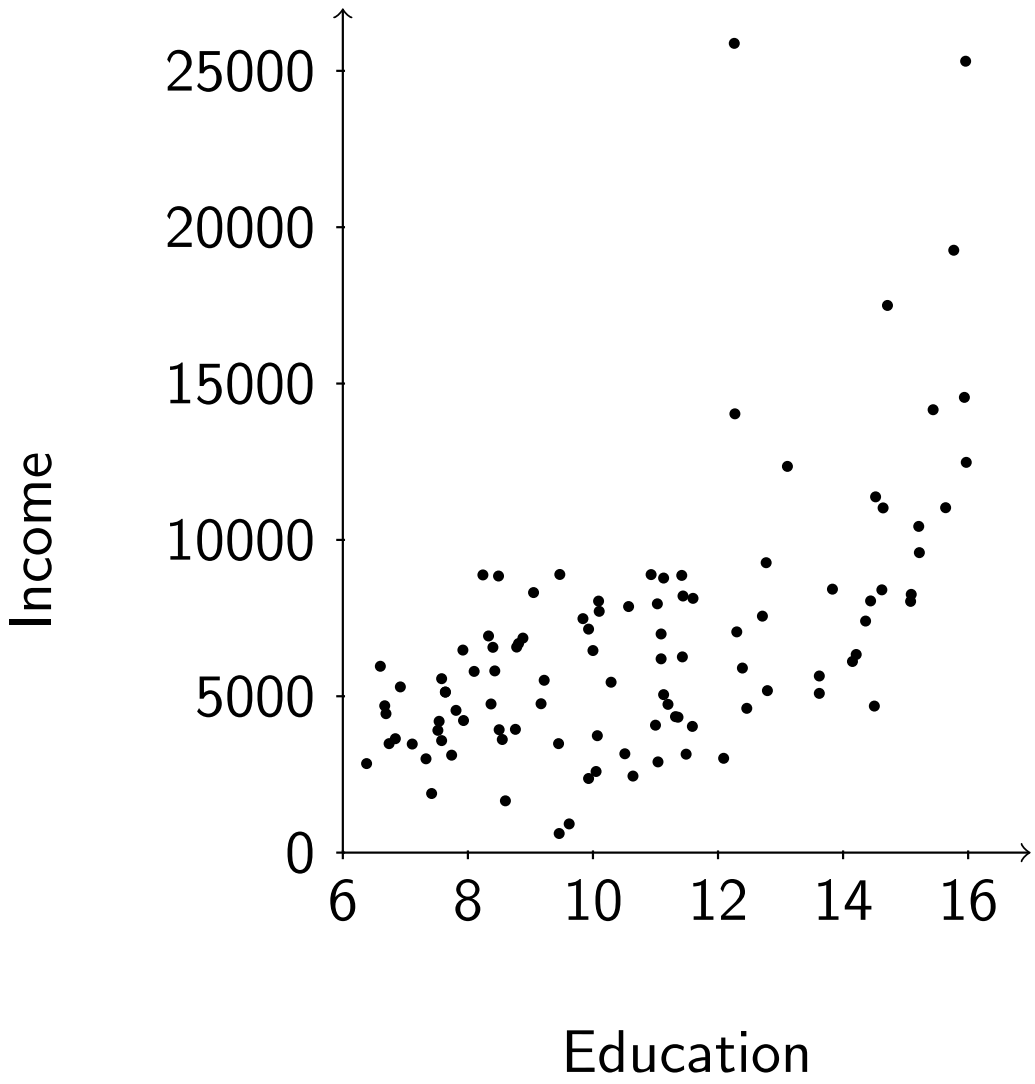
- Stacking pictures in Beamer is easy with overlays



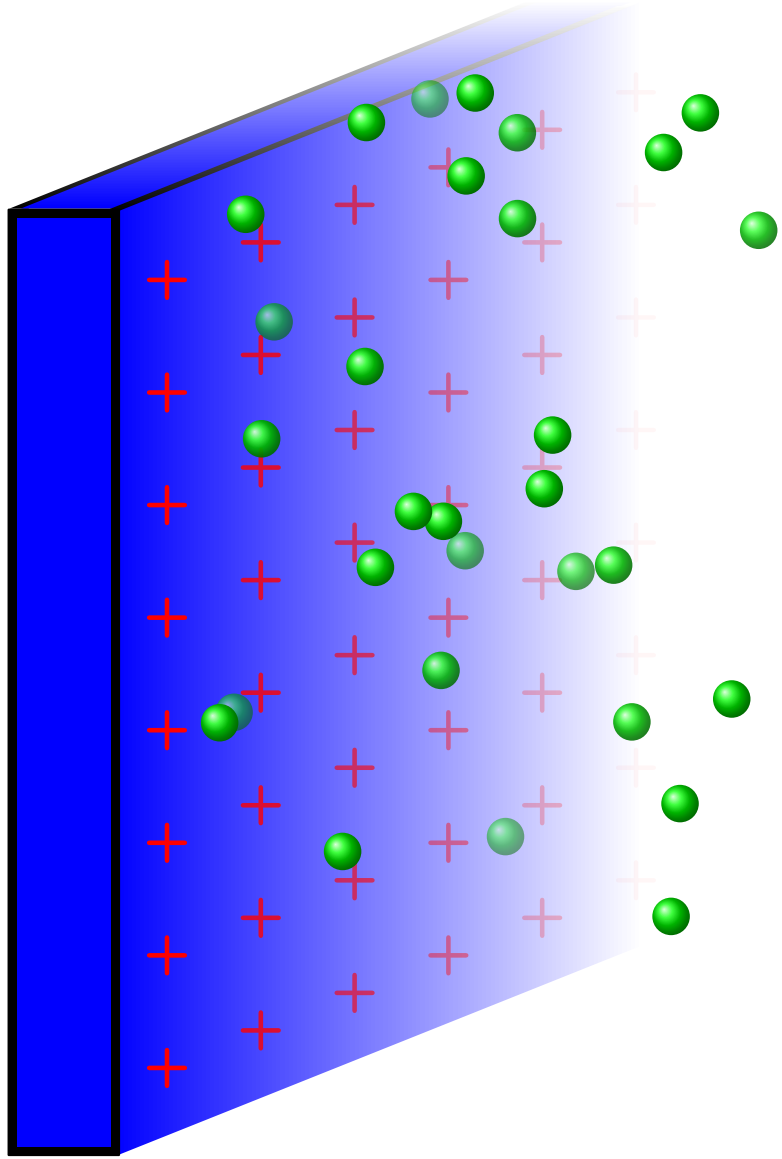
Plotting Functions



Plotting Data



Example: Membrane and Ions



Example: 3D Random Walk