

## Command Templates

- Basic path: `\drawing-command [options] path-specification;`
- Path specification: `(coordinate) path-component (coordinate);`
- Path Reusage `postaction={<basic drawing commands> or <decorate>}` When this option is given to any basic drawing commands below, the path is not immediately discarded and reused after the initial drawing command is finished.  
`preaction={<basic drawing commands> or <decorate>}` When this option is given to any basic drawing commands below, the path is used once before the initial drawing command is executed.

## Basic Drawing Commands

- `\path` : constructs a path

All following commands are in fact short forms for `\path` with one option or two:

- `\draw` : constructs and draws (`\stroke`) a path
- `\fill` : constructs and fills a path
- `\filldraw` : constructs, fills, and draws a path (in that order)
- `\shade` : constructs and shades a path
- `\shadedraw` : constructs, shades, and draws a path (in that order)
- `\coordinate` : label a coordinate
- `\node` : constructs a node

## Coordinate Specifications

- `(<x>,<y>)` : specifies the coordinate as a multiple of the current x- and y-vector (default: 1cm right and 1cm upwards)
- `(< $\theta specifies a coordinate in polar form with r being the vector length and  $\theta$  being the angle in degrees$`
- `+<coordinate specification>` : specifies a coordinate relative to the previous position but does not save the current position
- `++<coordinate specification>` : specifies a coordinate relative to the previous position

## Notes

1. Lengths can be with or without unit. If with a unit they are taken literally, if without they are multiples of the current 'x' or 'y' vector (as appropriate).
2. Relative coordinates are with respect to the last *\*saved\** position. Unless specified otherwise, the above all save their resultant position as the last saved position.
3. When a relative coordinate is used in a bezier curve specification the behaviour is slightly different. The second control point is taken relative to the final position of the curve, and the final point is taken relative to the initial one (or last saved position).

## Path Specifications

- `(coordinate) (coordinate)` : moves the "current point" from the first coordinate to the second
- `(coordinate) – (coordinate)` : draws a line from the first coordinate to the second
- `(coordinate) .. controls (control) and (control) .. (coordinate)` : draws a cubic bezier from the first coordinate to the second with the specified control points
- `(coordinate) to[options] (coordinate)` : draws a 'to path' from the first coordinate to the second; to paths can be extremely complicated
- `(coordinate) rectangle (coordinate)` : draws a rectangle with the coordinates as opposite corners
- `(coordinate) circle[options]` : draws a circle centred at the coordinate
- `(coordinate) arc[options]` : draws an arc starting at the coordinate
- `(coordinate) node[options] {text}` : adds a node at the coordinate
- `(coordinate) coordinate` : adds a coordinate label at the given coordinate

## Basic Options

- `draw[=<colour specification>] <sup>1</sup>` : draw the current path (with the given colour)
- `fill[=<colour specification>] <sup>1</sup>` : fill the current path (with the given colour)
- `<colour specification>` : set the colour for draw, fill, text (without explicitly enforcing those actions)
- `line width=<dimen>` : sets the line width
- `thin, thick, ultra thick` etc: presets for the line width

[1] The brackets indicate an optional part and must not be type in the code. I.e. use `draw=red` and not `draw[=red]`.