

## Lesson 20: Latex Picture environment

The basic syntax for the Latex picture environment is

```
\begin{picture}(width,height)(x-offset,y-offset)
...
  picture commands
...
\end{picture}
```

The picture environment allows you to create just about any kind of picture you want containing text, lines, arrows and circles. You tell LaTeX where to put things in the picture by specifying their coordinates. A coordinate is a number that may have a decimal point and a minus sign (-), for example, 5, 2.3, or -3.1416. A coordinate specifies a length in multiples of the unit length `\unitlength`, so if `\unitlength` has been set to 1cm, then the coordinate 2.54 specifies a length of 2.54 centimeters. You can change the value of `\unitlength` anywhere you want, using the `\setlength` command, but it should be set before the beginning of the picture environment since strange things may happen if you try changing it inside the picture environment.

A position is a pair of coordinates, such as (2.4,-5), specifying the point with x-coordinate = 2.4 and y-coordinate = -5. Coordinates are specified in the usual way with respect to an origin, which is normally at the lower-left corner of the picture. Note that when a position appears as an argument, it is not enclosed in braces; the parentheses serve to delimit the argument.

The picture environment has one mandatory argument, which specifies the size of the picture. The environment produces a rectangular box with width and height determined by this argument's two values.

The picture environment also has an optional position argument, following the size argument, that can change the origin. (Unlike ordinary optional arguments, this argument is not contained in square brackets.) The optional argument gives the coordinates of the point at the lower-left corner of the picture (thereby determining the origin). For example, if `\unitlength` has been set to 1mm, the command

```
\begin{picture}(100,200)(10,20)
```

produces a picture of width 100 millimeters and height 200 millimeters, whose lower-left corner is the point (10,20) and whose upper-right corner is therefore the point (110,220). Typically one initially omits the optional argument, leaving the origin at the lower-left corner. If you then want to modify your picture by shifting everything, you just add the appropriate optional argument.

The environment's mandatory argument determines the nominal size of the picture. This

need bear no relation to how large the picture really is; LaTeX will happily allow you to put things outside the picture, or even off the page. The picture's nominal size is used by TeX in determining how much room to leave for it.

Everything that appears in a picture is drawn by the `\put` command. The command

```
\put (11.3,-.3){obj}
```

puts the object specified by `obj` in the picture, with its reference point at coordinates (11.3,-.3). The reference points for various objects will be described below.

The `\put` command creates an LR box. You can put anything in the text argument of the `\put` command that you'd put into the argument of an `\mbox` (or related) command. When you do this, the reference point will be the lower left corner of the box.

See also

1. `\circle[\item ]{diameter}`

The `\circle` command produces a circle of the specified diameter. If the

2. `-form` of the command is used, LaTeX draws a solid circle.

3. `\dashbox{dash length}(width,height) [pos]{ ... }`

The `\dashbox` is like `\framebox` but has an extra argument which specifies the width of each dash. A dashed box looks best when the width and height are multiples of the dash length.

4. `\frame{ ... }`

The `\frame` command puts a rectangular frame around the object specified in the argument. The reference point is the bottom left corner of the frame. No extra space is put between the frame and the object

5. `\framebox[width] [position]{text}`

The `\framebox` command is exactly the same as the `\makebox` command, except that it puts a frame around the outside of the box that it creates.

The `framebox` command produces a rule of thickness `\fboxrule`, and leaves a space `\fboxsep` between the rule and the contents of the box.

In picture environment `\framebox(width,height) [position]{...}`

The `\framebox` command is exactly the same as the `\makebox` command, except that it puts a frame around the outside of the box that it creates.

6. `\line(x-slope,y-slope){length}`

The `\line` command draws a line of the specified length and slope.

The slope is determined by (x-slope,y-slope) which are signed integers of magnitude less than or equal to 6 and which have no common divisor except for plus or minus

one; For example (1,0) is a horizontal line; (1,1) gives a slope of 45 degrees; (0,1) is a vertical line; (-1,1) is 135 degrees.

The horizontal extent of the line is given by the length parameter, except in the case of a vertical line in which case length specifies the vertical height.

7. `\linethickness{dimension}`

Declares the thickness of horizontal and vertical lines in a picture environment to be dimension, which must be a positive length. It does not affect the thickness of slanted lines and circles, or the quarter circles drawn by `\oval` to form the corners of an oval.

8. `\makebox[width] [position]{text}`

The `\makebox` command creates a box to contain the specified text. The width of the box may be specified by the optional width argument. The text in the box is set in LR mode.

If the optional width argument is used, the optional position argument may also be used to specify the position of text within the box as follows:

- (a) l - flushleft
- (b) r - flushright
- (c) s - interword space adjusted so text fills box exactly

The default horizontal and vertical positioning (if the second optional argument is omitted) is to center text in the box.

Note that text may overflow the box, in which case it is typeset on top of the adjacent text. While this is generally undesirable, it can also be used to create special effects.

In picture environment `\makebox(width,height) [position]{ ... }`

The `\makebox` command for the picture environment is similar to the normal `\makebox` command except that you must specify a width and height (note that these are in parentheses) in multiples of `\unitlength`.

The optional argument, position, specifies the quadrant that your text appears in. You may select up to two of the following:

- (a) t Moves the item to the top of the rectangle
- (b) b Moves the item to the bottom
- (c) l Moves the item to the left
- (d) r Moves the item to the right

The default horizontal and vertical positioning is to center text in the box.

9. `\multiput(x coord,y coord)(delta x,delta y){number of copies}{object}`

The `\multiput` command can be used when you are putting the same object in a regular pattern across a picture.

10. `\oval(width,height) [portion]`

The `\oval` command produces a rectangle with rounded corners. The optional argument, portion, allows you to select part of the oval.

- (a) t Selects the top portion
- (b) b Selects the bottom portion
- (c) r Selects the right portion
- (d) l Selects the left portion

11. `\put(x-coord,y-coord){object}`

The `\put` command places the object specified by the mandatory argument at the given coordinates.

The coordinates are in units of `\unitlength`

Some of the object items you might place are

- (a) line
- (b) vector
- (c) circle
- (d) oval

12. `\shortstack[position]{ ... \\ ... \\ ... }`

The `\shortstack` command produces a stack of objects. The valid positions are:

- (a) r Moves the objects to the right of the stack
- (b) l Moves the objects to the left of the stack
- (c) c Moves the objects to the center of the stack (default)

13. `\unitlength`

The `\unitlength` command is a Length command. It defines the units used in the Picture Environment

The default value is 1 point (approximately 1/72 inch) but it may be changed using the `\setlength` command.

14. `\vector(x-slope,y-slope){length}`

The `\vector` command draws an arrow of the specified length and slope with the arrow head at the opposite end of the line from the reference point (see `\put`).

The slope is determined by (x-slope,y-slope) which are signed integers of magnitude less than or equal to 4 and which have no common divisor except for plus or minus one; For example (1,0) is a horizontal arrow pointing right; (1,1) gives a slope of 45 degrees; (0,1) is a vertical arrow pointing up; (-1,1) is at 135 degrees.

The horizontal extent of the arrow is given by the length parameter, except in the case of a vertical arrow in which case length specifies the vertical height.

## Lengths in Latex

A length is a measure of distance. Many LaTeX commands take a length as an argument. A length of one inch is specified by typing `1in` or, to specify the same length using metric units, by typing `2.54cm`.

Units LaTeX knows about the following common units.

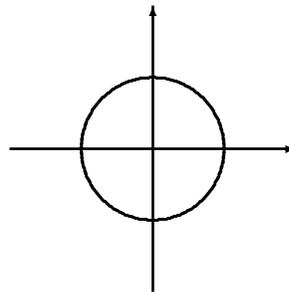
1. `in` - inches
2. `mm` - millimeters
3. `cm` - centimeters
4. `pt` - points (about  $1/72$  inch)
5. `em` - approximately the width of an “M” in the current font
6. `ex` - approximately the height of an “x” in the current font

Lengths may also be negative, for example, `-1.5em`. Note that the number 0 by itself is not a length; it must be specified as `0in` or `0pt`, for example.

### Example 1: An axis and circle

Note that to make good circles you should use the `curves.sty` style file.

```
\setlength{\unitlength}{.5in}
\begin{picture}(10,10)(-5,5)
\linethickness{1pt}
\put(-2.5,0){\vector(1,0){5}}
\put(0,-2.5){\vector(0,1){5}}
\put(0,0){\circle{2}}
\end{picture}
```

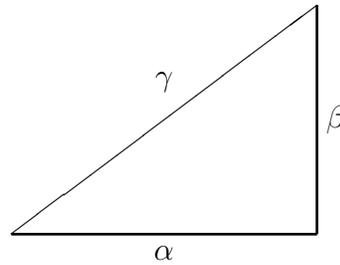


## Example 2: A right triangle

```

\setlength{\unitlength}{.4in}
\begin{picture}(7,5)(0,0)
\linethickness{1pt}
\put(0,0){\line(1,0){4}}
\put(4,0){\line(0,1){3}}
\put(0,0){\line(4,3){4}}
\put(2,-.25){\makebox(0,0){$\alpha$}}
\put(4.25,1.5){\makebox(0,0){$\beta$}}
\put(2,2){\makebox(0,0){$\gamma$}}
\end{picture}

```



## Example 3: A discontinuous function

```

\setlength{\unitlength}{.01in}%.025in}
\begin{picture}(200,75)
\put(0,25){\vector(1,0){200}}
\put(25,0){\vector(0,1){75}}
\put(75,22){\line(0,1){6}}
\put(125,22){\line(0,1){6}}
\put(22,50){\line(1,0){6}}
\thicklines
\put(25,25){\line(1,0){50}}
\put(75,50){\line(1,0){50}}
\put(125,25){\line(1,0){72}}
\put(17,50){\makebox(0,0){$1$}}
\put(75,13){\makebox(0,0)[b]{$\pi$}}
\put(125,13){\makebox(0,0)[b]{$2\pi$}}
\put(195,13){\makebox(0,0)[b]{$t$}}
\put(175,60){\makebox(0,0){$g(t)$}}
\end{picture}

```

