

The Graphics window

The Graphics window is the drawing surface for the turtles.

BUTTON? (BUTTONP)

Reports the state of a mouse button.

Syntax

```
BUTTON?  
(BUTTON? button)
```

Description

BUTTON? reports the state of a mouse button. BUTTON? reports TRUE when the left button is depressed or FALSE when the button is not depressed. Is used with an additional input, the input is 1 for the left button, 2 for the middle button, and 3 for the right button.

See also [MOUSE](#).

Example

```
TO CHASE  
  SETPOS MOUSE  
  IF BUTTON? [STOP]  
  CHASE  
END  
CHASE
```

CLEAN

Erases the graphics pane.

Syntax

```
CLEAN
```

Description

CLEAN erases the current Graphics window but does not affect the heading or position of the turtle. See also [CLEARSCREEN](#) and [DRAW](#).

Example

```
HOME  
SETPC "RED  
SETH 20  
FD 50  
CLEAN  
FD 50
```

CLEARSCREEN (CS)

Erases the graphics pane and homes the turtle.

Syntax

```
CLEARSCREEN
```

Description

CLEARSCREEN erases the current Graphics window, deletes all control widgets, returns the turtle to the center of the window, and sets the turtle's heading to 0. CLEARSCREEN does not affect the pen state or window colors. See also [CLEAN](#) and [DRAW](#).

Example

```
HOME  
SETPC "RED  
SETH 20  
FD 50  
CLEARSCREEN  
FD 50
```

DRAW

Clears the current graphics window and resets all attached turtles.

Syntax

```
DRAW
```

Description

DRAW prepares the current Graphics window for drawing by doing the following:

- Clears the window.
- Deletes all control widgets.
- Resets all attached turtles to their default values.
- Sets turtle 0 to be the active turtle.
- Homes the turtle.
- Shows the turtle.
- Puts the pen down.
- Sets the window to [WRAP](#) mode.

See also [CLEARSCREEN](#) and [CLEAN](#).

Example

```
TELL 0 HOME
```

```
RT 22 FD 10000  
DRAW
```

EXTENT

Reports the extent of the current Graphics window.

Syntax

```
EXTENT
```

Description

EXTENT reports a list containing the X and Y extents of the current Graphics window, based on the extent settings of the first active turtle. In the default state, where one turtle step corresponds to one screen pixel, the extents are both zero.

For a detailed explanation of the window extent, see [SETEXTENT](#).

See also [SETEXTENT](#), [WRAP](#), and [FENCE](#).

Example

```
EXTENT  
Result: [0 0]
```

FENCE

Fences all turtles inside the drawing borders.

Syntax

```
FENCE
```

Description

FENCE prevents the turtle from moving beyond the edge of the Graphics drawing borders. If you try to move the turtle outside the borders, it moves to the drawing border and stops.

See also [WRAP](#) and [WINDOW](#).

Example

```
HOME  
SETH 20  
FENCE  
FD 1000
```

GRID

Sets the size of the graphics grid.

Syntax

```
GRID size
```

Description

GRID sets the horizontal and vertical scales of the Graphics window grid to its input.

See also [GRIDON](#) and [GRIDOFF](#).

Example

```
GRIDON  
GRID 50  
GRID 100
```

GRIDOFF

Hides the grid of the Graphics window.

Syntax

```
GRIDOFF
```

Description

GRIDOFF makes a visible grid in the Graphics window invisible.

See also [GRIDON](#) and [GRID](#).

Example

```
GRIDON  
GRIDOFF
```

GRIDON

Displays the grid of the Graphics window.

Syntax

```
GRIDON  
(GRIDON [horizontal vertical])
```

Description

GRIDON makes a grid visible in the Graphics window. By default, the scale of the grid is 100 pixels on each axis.

The scale of the grid may be changed by supplying GRIDON with a list containing the desired horizontal and vertical scales for the grid and putting GRIDON and its argument in parentheses.

See also [GRIDOFF](#) and [GRID](#).

Example

```
(GRIDON [150 50])  
GRIDON
```

LOADPIC

Loads a picture as background picture.

Syntax

```
LOADPIC filename.ext  
(LOADPIC filename.ext TRUE)
```

Description

LOADPIC loads the file from the cloud to the current graphics canvas. The file is loaded as a bit map that fills the current viewport. If no extension is given to the filename, the picture format that is loaded is the format specified in the built-in variable `:PICTURE.FORMAT`. If the file has a different extension, then the extension must be specified in the LOADPIC command.

If the last input to LOADPIC is TRUE, LOADPIC loads multiple copies of the image instead and fills the entire graphics screen with these copies (it tiles the images).

MOUSE

Reports the position of the mouse cursor.

Syntax

```
MOUSE
```

Description

MOUSE reports the coordinates of the position of the mouse pointer as a list of two numbers representing the x coordinate and the y coordinate. The coordinates are relative to the center of the Graphics window. See also [BUTTON?](#).

Example

```
TO CHASE  
  SETPOS MOUSE  
  IF BUTTON? [STOP]
```

```
CHASE
END
CHASE
```

MOUSESHAPE

Reports the shape of the mouse cursor.

Syntax

```
MOUSESHAPE
```

Description

MOUSESHAPE reports the number of the current mouse pointer shape as defined with the [SETMOUSESHAPE](#) command. On startup, the mouse shape is 1.

Example

```
MOUSESHAPE
```

SAVEPIC

Saves the contents of the Graphics window.

Syntax

```
SAVEPIC filename.ext
SAVEPIC type-descriptors
(SAVEPIC)
(SAVEPIC filename.ext "windowname")
```

Description

SAVEPIC saves the contents of the current Graphics window into a bitmap file on disk. SAVEPIC uses the boundaries of the current drawing area. If no extension is given to the filename, the picture format that is saved is the format specified in the built-in variable [:PICTURE.FORMAT](#). If you want a different format than the default, then the extension must be specified in the SAVEPIC command. SAVEPIC can save into several different formats; see [:PICTURE.FORMAT](#) for a list of supported formats.

SAVEPIC displays a File Open dialog if it is called without any inputs, or if the input is a list of file type descriptors. For a description of file type descriptors, see the [OPEN](#) command. Clicking the Cancel button in a File Save dialog causes the SAVEPIC command to return.

Example

```
SAVEPIC "Sample"
```

SETEXTENT

Set the extent of the current Graphics window.

Syntax

```
SETEXTENT extent  
SETEXTENT [xExtent yExtent]  
(SETEXTENT)
```

Description

SETEXTENT defines the coordinate mapping mode in terms of turtle steps. The default is an extent value of zero, which means the coordinate system has an aspect ratio of 1:1 regardless of the window size. Painting outside the window in [WINDOW](#) mode is possible and the drawings become visible when the window is made larger.

If a different extent is defined, the coordinate system is mapped to the size given regardless of the window size. If an extent value of 200 is set, the coordinates range from -200 to +200. Scaling is performed automatically to fit within the window. It is possible to set different extents for the X and Y coordinates by using a two-element list as input. (SETEXTENT) resets the extent to 1:1 mapping.

Note that each individual turtle, control, or bitmap may have its own screen extent. Thus the lines drawn by different turtles for the same number of turtle steps may be of different lengths if their extents are not the same.

See also [EXTENT](#), [SETORIGIN](#) and [ORIGIN](#).

Example

```
ASK 0 [SETEXTENT 100 PENUP SETX -25 PENDOWN]  
ASK 1 [SETEXTENT 50 PENUP SETX 25 PENDOWN]  
TELL [0 1]  
ST FORWARD 30
```

SETMOUSESHAPE

Sets the shape of the mouse cursor.

Syntax

```
SETMOUSESHAPE number  
(SETMOUSESHAPE)
```

Description

SETMOUSESHAPE changes the shape of the mouse cursor when it is within the Graphics window. Its input is a number representing one of 23 available mouse shapes. When Logo starts, the mouse shape is 1. If SETMOUSESHAPE is enclosed in parentheses without inputs, shape 1 is used.

[MOUSESHAPE](#) reports the number of the current mouse shape.

Example

```
SETMOUSESHAPE 20
```

WINDOW

Removes the boundary for turtle movements.

Syntax

```
WINDOW
```

Description

`WINDOW` removes the boundaries from the turtle's field of movement. If the turtle moves beyond the borders of the Graphics drawing borders, it continues to move, but cannot be seen. The Graphics window becomes a small window overlooking the plane on which the turtle can travel.

See also [FENCE](#) and [WRAP](#).

Example

```
HOME  
SETH 20  
WINDOW  
FD 1500
```

WRAP

Lets the turtles wrap inside the window.

Syntax

```
WRAP
```

Description

`WRAP` maintains the turtle inside the Graphics drawing borders no matter how large a movement command is given. Any time the turtle moves off the window border, it wraps around the window and reappears on the opposite edge. When Logo starts up, the default window state is `WRAP`.

See also [FENCE](#) and [WINDOW](#).

Example

```
HOME  
SETH 20  
WRAP  
FD 1500
```