

Math on a Sphere: Common Commands (Page 1)

Hints:

- The green triangle in the Sphere Window represents a turtle. We create drawings by moving the “turtle” around.
- In these command sheets, words in the `Courier` font or the **bold Courier font** are commands.
- Some commands have spaces in their names, and others don't, so be careful.
- For commands like `forward`, you don't need to use the `#` symbol. The `#` symbol just indicates that you need to choose a number, such as `forward 10`
- Be careful about the types of brackets you use! Sometimes you will need to use square brackets `[]` and other times you will need to use curly brackets `{}`. For example, the `towards`, `setposition` and `distanceto` commands all use square brackets `[]`. On the other hand, the `repeat` command uses curly brackets `{}`.

forward # (abbreviation: **fd**)

Tells the turtle to walk forward a specific number of steps

To move the turtle forward 40 steps: forward 40 or fd 40

back # (abbreviation: **bk**)

Tells the turtle to walk backward a specific number of steps

To move the turtle backward 100 steps: back 100 or bk 100

right # (abbreviation: **rt**)

Tells the turtle to turn to the right a specific number of degrees

To turn the turtle to the right by 90 degrees: right 90 or rt 90

left # (abbreviation: **lt**)

Tells the turtle to turn to the left a specific number of degrees

To turn the turtle to the left by 30 degrees: left 30 or lt 30

Math on a Sphere: Common Commands (Page 2)

penup

Tells the turtle to lift its pen off the ground, so that it can move without drawing a line. (It may help to imagine that the turtle has a pen attached to its tail. When you tell the turtle `penup` it lifts up its tail, so it can walk without the pen dragging on the ground.)

pendown

Tells the turtle to put its pen back down, so that when it moves it draws a line. (It may help to imagine that the turtle has a pen attached to its tail. When you tell the turtle `pendown` it puts its tail back on the ground.)

setspeed #

Tells the turtle how fast to walk. Note that the turtle's maximum speed is 50, so a number larger than 50 will still result in a speed of 50.

To set the speed to 20: `setspeed 20`

set pensize #

Tells the turtle to use a different thickness pen, so it makes either a thinner or thicker line.

Note: turtles cannot make pens thinner than 0.1, so any number smaller than 0.1 results in a thickness of 0.1. Also, turtles don't like pens thicker than 90, so if you try to use a number bigger than 90 the turtle gets mad and won't draw anything.

To set the size of the pen to 4: `set pensize 4`

set color #

Changes the color of the lines on the sphere. You can specify the color with a number or the name of a color. A color map and inputs for common colors are given on the next page.

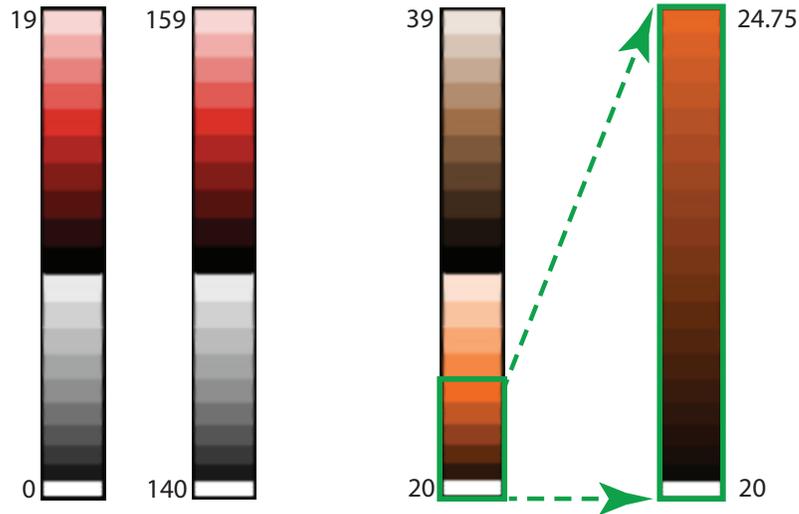
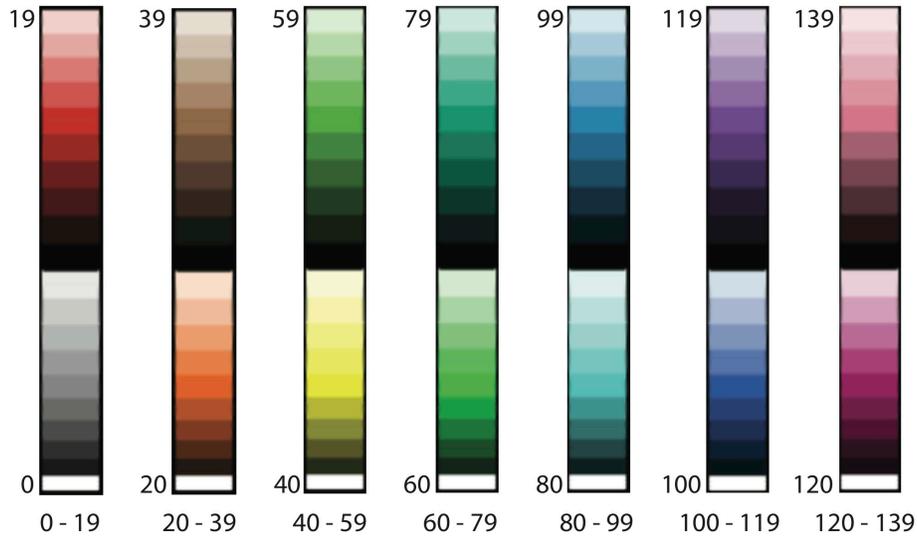
To set the color to 125 (magenta): `set color 125`

To set the color to cyan: `set color cyan`

ca

Clears your drawing. If you are giving the turtle commands through the Command Line Window and you want to erase that lines you've already drawn, use this command.

Math on a Sphere: Color Map



Colors start repeating at 140

Color inputs don't have to be whole numbers

set color inputs for common colors			
0	black	25	orange
5	gray	35	brown
9.9	white	45	yellow
15	red	55	green
		65	lime
		75	turquoise
		85	cyan
		95	sky
		105	blue
		115	violet
		125	magenta
		135	pink

Math on a Sphere: Advanced Commands and Syntax (Page 1)

Commenting code

Double slashes // indicate comments. They tell the turtle that you're going to write in human language, not turtle language. The turtle only cares about turtle language, so the turtle ignores any words that come after double slashes. Programmers use comments to organize their code and remember what they did.

Example: // This is Lizzie's code for drawing a soccer ball (5/5/2012)

setheading

Tells the turtle what heading (angle) to point towards. It may help to think of the turtle's body as a compass and this command as a way to set the compass heading. More information about this can be found in the topic page called "Headings and Positions".

Note: you can use any positive number for the heading. (For example, you can choose a heading of 450 degrees. The result will be the same as it would be if you'd chosen 90 degrees.) To point the turtle in the initial direction of the "north" pole, use a heading of 0 degrees.

To set the heading of the turtle to 45 degrees: `setheading 45`

getheading

Prints the turtle's heading in the command line window. *This command only works in the Command Line Window – it does not work in the Editor Window.* For more information about headings, see `setheading`, above.

To find the turtle's heading: `getheading`

towards [x,y]

Turns the turtle so that it points toward the point [x,y]. (This command turns the turtle, so it changes the turtle's heading.)

To turn the turtle toward the point [0,10]: `towards [0,10]`

distanceto [x,y]

To calculate the distance to the point [0,10] on the sphere: `distanceto [0,10]`

Calculates and prints the distance from the turtle's current position to the supplied point [x,y]. *This command only works in the command line window – it does not work in the Editor Window.*

Math on a Sphere: Advanced Commands and Syntax (Page 2)

setposition [x,y]

Drags the turtle to a new position [x,y]. If you want the turtle to jump to the new position instead, you must use the `penup` command before the `setposition` command. *This command affects the heading.*

To drag the turtle to the position [0,3]: `setposition [0,3]`

- **Coordinates system:** A pair of coordinates [x,y] is used to describe the turtle's position on the surface of the sphere. The origin [0,0] is at the point along the equatorial line closest to the viewer in the "Sphere Window". *x* is measured in degrees from the equator to the north and *y* is measured from the origin along the equator in an eastward direction.
- **Important!** You will usually want to use `setposition` in combination with the `setheading` command, because `setposition` changes the heading!

getposition

Prints the turtle's position in the command line window. *This command only works in the command line window – it does not work in the Editor Window.* For more information about positions, see `setposition`, above.

To find the turtle's position: `getposition`

repeat # { write some instructions here }

The `repeat` command tells the turtle to repeat a set of commands a specific number of times. Make sure you use curly brackets and tell the turtle how many times to repeat the commands.

In the following example, we are telling the turtle to take 41 steps forward, turn to the right 117 degrees, and then repeat these two commands two more times, for a total of 3 times.

Example: `repeat 3 {forward 41 right 117}`

You can also type your code in a different way, by placing different items on different lines, as shown below:

```
repeat 3 {  
    forward 41  
    right 117  
}
```

