

## Logotacular - reference

Logotacular is a colourful app to help you learn 'Logo', a programming language designed to assist in learning geometry and programming.

The most basic commands in Logotacular are the 'go forward' command and the 'turn right' command. You use these commands to control a small on-screen 'turtle', which draws a colourful path as it moves. You can change the background colour, line colour and the thickness of the lines drawn, as well as raising/lowering the pen.

Loops can be programmed by use of the 'repeat' command which tells the turtle to execute a set of commands a specific number of times.

More advanced users can use variables are used to tell the turtle the value assigned to a letter, and functions (or 'procedures') are used to name a set of commands so that you can easily execute them later.

I no longer support or work on the web application version of Logotacular since the iPad app has gained more traction. The screenshots below are taken from the iPad app, but I keep the legacy web application in case anyone wants to use it and doesn't have an iPad.

### Basic commands

Command	Example	What it does
fd, bk	fd 100 bk 100	Move forwards or backwards 100 units
rt, lt	rt 90 lt 90	Turn right or left 90 degrees
home	home	Reset the turtle's position to where it started
penup/pu	penup	Lift the pen off the paper
pendown/pd	pendown	Place the pen back down on the paper
color/colour	color yellow / color 17	Change the color of the pen
bg	bg blue / bg 8	Change the color of the background
thick	thick 8	Change the thickness of the lines (from 1 to 20)
setxy	setxy 100 100	Move the turtle to the co-ordinates (100, 100)
random	color random fd random	Choose a random color and then go forward a random amount (between 0 and 100)

## Repeating, functions and advanced commands

Command	Example	What it does
<code>rpt 10 [ ... ]</code>	<code>rpt 10 [ fd 100 rt 36 ]</code>	Repeat the commands inside the square brackets 10 times
<code>make "varname</code>	<code>make "num 16</code>	Set the variable 'num' to 16
<code>label</code>	<code>label 'here</code> <code>label :num</code>	Write the word 'here' or the value of the variable 'num' to the screen, at the turtle's position
<code>:varname</code>	<code>fd :num</code>	Use the variable 'num' (for example go forward 16 units)
<code>to procname</code> <code>...</code> <code>end</code>	<code>to drawsquare</code> <code>  rpt 4 [</code> <code>    fd 150</code> <code>    rt 90</code> <code>  ]</code> <code>end</code> <code>drawsquare</code>	Define a function called 'drawsquare', which takes no input, and then call it

• • • •

## Program control

Command	Example	What it does
<code>stop</code>	<code>to drawpoly :n</code> <code>if :n = 0 [stop]</code> <code>...</code> <code>end</code> <code>drawpoly 10</code>	Stop a function or loop from finishing execution
<code>if ... [ ... ]</code>	<code>if (:a &lt; 1) [fd 100]</code>	Execute the statements in the square brackets only when 'a' is less than one. You can use the following operators: = (equal to) < (less than) > (greater than) <= (less than or equal to) >= (greater than or equal to)
<code>ifelse ... [ ... ] [ ... ]</code>	<code>ifelse (:a &lt;= 1) [fd 100] [fd -100]</code>	Execute the statements in the first square brackets only when 'a' is less than or equal to one, and the statements in the second square brackets otherwise.

• • • •

## Colors you can use

terracotta/dkred/0	red/1	fuchsia/2	pink/3
wisteria/4	purple/violet/5	indigo/6	midnight/7
blue/8	lightblue/ltblue/9	cyan/10	sage/11
emerald/12	turquoise/13	darkgreen/dkgreen/14	green/15
applegreen/16	yellow/17	ltorange/lightorg/ ltorg/lightorange/18	orange/org/ carrot/19
dkorange/darkorg/ darkorange/dkorg/20	brown/21	chocolate/22	black/23
darkgray/dkgray/ darkgrey/dkgrey/24	gray/grey/25	lightgrey/ltgrey/ lightgray/ltgray/26	white/27

• • • •