

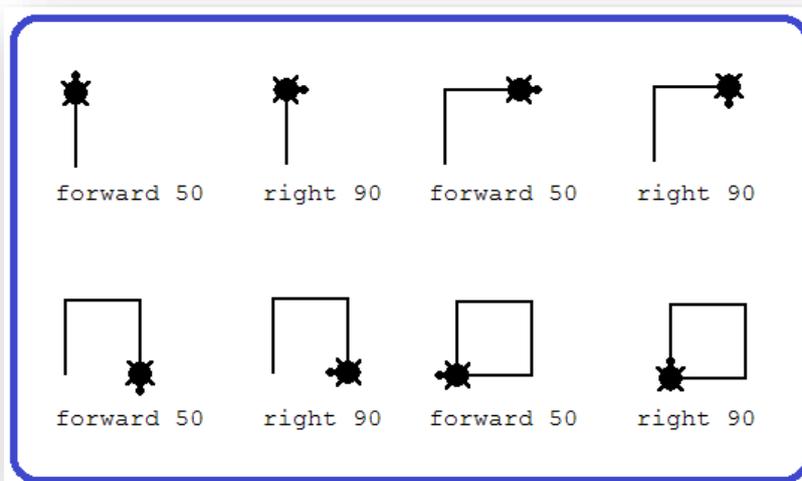
# EVANGELOS C. ZIOULAS

IT Teacher



## CHAPTER 6

## TURTLE'S GEOMETRY



## GENERAL FEATURES

One of the most interesting features of Logo is the **microworld** which is also called **Turtle's geometry**.

In Microworlds, there is a turtle that lives in the screen and corresponds to our commands (primitives available in the vocabulary of Logo, or procedures defined by the user).



The effects of Logo commands on the turtle are visible on the screen. The turtle can design geometrical shapes and other patterns in the screen by depicting its trace as it moves on the page.

The most **important instructions of Logo** are those which change the turtle's state as well as modify the graphics in the screen.

TURTLE'S INSTRUCTIONS	EXPLANATION
<b>forward</b> number or <b>fd</b> number	The turtle moves forward as many steps as the number indicates
<b>back</b> number or <b>bk</b> number	The turtle moves backwards as many steps as the number indicates
<b>right</b> number or <b>rt</b> number	It turns the turtle to the right as many degrees as the number indicates
<b>left</b> number or <b>lt</b> number	It turns the turtle to the left as many degrees as the number indicates
<b>pd</b>	It puts down the pen of the current turtle. The turtle then leaves a trace when it moves but not when it is dragged
<b>pu</b>	It lifts up the pen of the current turtle, so it will not leave a trace when it moves
<b>setpensize</b> number	It sets the turtle's pen size which determines the thickness of the lines it will draw (1..30)
<b>setcolor</b> number or name <b>setc</b> number or name	It sets the color of the turtle's pen. If the turtle has its original shape, it changes color to show the pen color.
<b>setheading</b> number <b>seth</b> number	It sets the turtle's heading to the specified direction (it is expressed in degrees 0..360)
<b>setopacity</b> number	It sets the opacity of the turtle and its pen with a number from 0 (transparent) to 100 (opaque)
<b>setsh</b> number or list or name	It gives a shape or a set of shapes to the turtle (maximum 128), so when moving it cycles through the list of shapes
<b>setpos</b> [x y]	It moves the turtle to the designated x, y coordinates. The center point of the page is [0 0]
<b>setx</b> number	It sets the x coordinate of the current turtle. The y coordinate remains unchanged.
<b>sety</b> number	It sets the y coordinate of the current turtle. The x coordinate remains unchanged.
<b>glide</b> distance speed	It makes the turtle glide over a particular distance (0..9999) with a particular speed (0..99)
<b>fill</b>	It fills a closed area with the turtle's pen color. If the area is not perfectly closed, the paint "leaks" onto the entire page.
<b>stamp</b>	It stamps a copy of the turtle on the background. The pen does not have to be placed down to stamp
<b>towards</b> turtle	It sets the heading of the current turtle to aim towards the turtle whose name is given as input.

GRAPHICS INSTRUCTIONS	EXPLANATION
<b>cg</b>	It clears the graphics on the page and returns the current turtle to its home position, pointing up
<b>clean</b>	It clears the graphics without changing any turtle's position.
<b>setbg number</b>	It sets the background color for the page. The input can be the name of a color or a number
<b>snapshot</b>	It takes a snapshot of the background. Next time a "restore" command is used, the background will be restored to that snapshot
<b>restore</b>	It restores the background to the way it was the last time a snapshot command was issued. The turtle's position does not change and everything else remains intact
<b>freezebg</b>	It freezes the background graphics in their current state. We can still draw over the background and erase the new drawings, but the frozen background won't be erased
<b>unfreezebg</b>	It unfreezes the background that was frozen by "freezebg"
<b>clearshape number or list</b>	It clears the shape or shapes listed in the Public Shapes Tab. The input must be a number or a list of numbers between 1 and 128

REPORTERS	EXPLANATION
<b>bg</b>	It reports a number representing the color of the background.
<b>color</b>	It reports the turtle's color as a number.
<b>heading</b>	It reports the current turtle's heading in degrees. (0 for North, 90 for East, 180 for South and 270 for West)
<b>pensize</b>	It reports a number representing the pen size of the current turtle (the original size is 1 and the maximum is 30)
<b>opacity</b>	It returns the opacity of the current turtle's pen and shape. The value is a percentage.
<b>pos</b>	It reports the position of the turtle as a list of two numbers (coordinates). The position at the center of the page is [0 0]
<b>size</b>	It reports the size of the current turtle
<b>shape</b>	It reports the shape name or number, or the list of shape names or numbers of the current turtle.
<b>distance turtle</b>	It reports the distance between the current turtle and the turtle indicated
<b>who</b>	It reports the name of the current turtle
<b>sender</b>	It reports the name of the turtle who has made the last "broadcast" or "tell" command

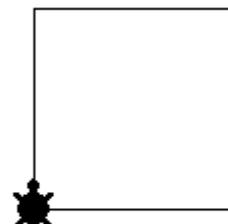
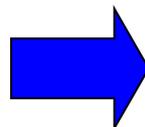
MORE INSTRUCTIONS	EXPLANATION
<code>newturtle όνομα</code>	It creates a new turtle with the name indicated. The new turtle appears at the position [0 0] and is hidden
<code>talkto list of turtles</code> <code>tto list of turtles</code>	It makes many turtles do the same thing at the same time
<code>broadcast word</code>	It sends a message to all the turtles on the current page to hear. It triggers the instruction found in their OnMessage field
<code>tell list of turtles</code>	It sends a message to particular turtles on the current page to hear. It triggers the instruction found in their OnMessage field
<code>everyone [instruction list]</code>	It makes all the turtles on the current page run the list of instructions inside the square brackets
<code>home</code>	It moves the turtle to the center of the page coordinates [0 0], pointing up.
<code>ht</code>	It hides the current turtle on the screen
<code>st</code>	It shows the current turtle
<code>inback</code>	It puts the current turtle behind all the other turtles on the page.
<code>infront</code>	It brings the current turtle in front of all the other turtles on the page. A newly created turtle is always in front of the others.
<code>clone turtle</code>	It creates a copy of the named turtle. The new turtle takes the first available name on the current page
<code>clickon</code>	It clicks on the current turtle, stimulating the instructions included in its OnClick field.
<code>clickoff</code>	It clicks on the current turtle, stopping the execution of the instructions that are running in its OnClick field

**Example:** To draw a **square** that has a side of 100 steps:

```
pd
forward 100 right 90
forward 100 right 90
forward 100 right 90
forward 100 right 90
```

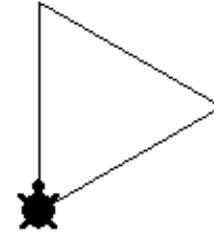
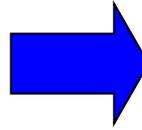
Alternatively

```
pd
fd 100 rt 90
fd 100 rt 90
fd 100 rt 90
fd 100 rt 90
```



**Example:** To draw an **equilateral triangle** which has a side of 100 steps:

```
pd  
forward 100 right 120  
forward 100 right 120  
forward 100 right 120
```



## THE REPEAT COMMAND

If we study carefully the previous examples, we can easily observe that there are sets of commands that are repeated for a number of times e.g. `forward 100 right 90` (square) or `forward 100 right 120` (triangle).

As a shortcut, we can use the **repeat** command, that is a primitive which accepts a **list of instructions inside square brackets [ ]** and specifies the number of iterations to be executed e.g.

To create a square with 100 steps size



```
repeat 4 [forward 100 right 90]
```

To create a triangle with 100 steps size



```
repeat 3 [forward 100 right 120]
```

We notice that the turtle either draws a square or a triangle, and it **rotates constantly 360 degrees** in total.

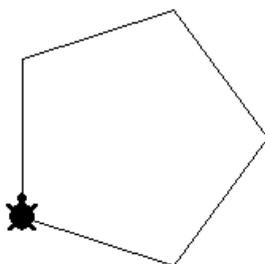
Therefore, for a turtle to make a **complete rotation** in order to draw a **closed polygon**, it needs to be rotated 360 degrees in total.

## FULL PATH THEOREM

A turtle must totally rotate **360 degrees** in order to complete its route around a closed shape as well as return to its initial position and direction.

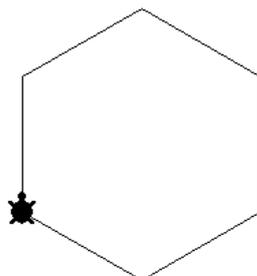
Examples:

```
repeat 5 [fd 100 rt 72]
```



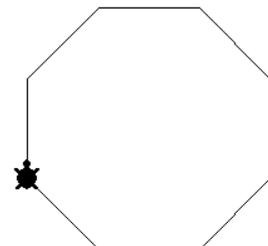
pentagon

```
repeat 6 [fd 100 rt 60]
```



hexagon

```
repeat 8 [fd 100 rt 45]
```



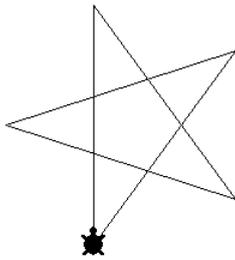
octagon

## THE DESIGN OF A STAR

To draw a star (with at least 5 apices), the turtle should make at least 2 complete rotations of 360 degrees each, that means **720 degrees**.

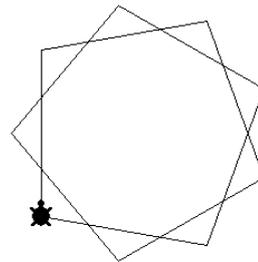
```
repeat 5 [fd 100 rt 144]
```

because  $5 * 144 = 720$



```
repeat 9 [fd 100 rt 80]
```

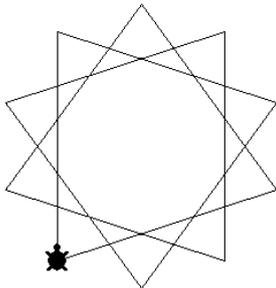
because  $9 * 80 = 720$



To draw stars with a greater number of apices, we should cover more than 2 complete rotations. For example, 3 complete rotations are equal to  $3 * 360 = 1080$  degrees

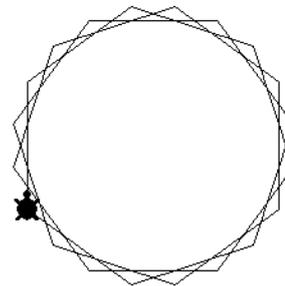
```
repeat 10 [fd 100 rt 108]
```

because  $10 * 108 = 1080$



```
repeat 20 [fd 100 rt 54]
```

because  $20 * 54 = 1080$



## GENERALIZATION OF FULL PATH THEOREM

A turtle must rotate totally 360 degrees or a **multiple of 360 degrees** (e.g. 720, 1080 etc.) in order to complete its route around a closed shape as well as return to its initial position and direction.

We can design more complex and impressive shapes when we draw the **same shape repeatedly** in conjunction with what the full path theorem defines (full rotation of 360 degrees).

(nested structure)

```
repeat 5 [repeat 4 [fd 100 de 90] rt 72]
```

Square design =  $4 * 90$  degrees

A complete rotation =  $5 * 72 = 360$  degrees

