TURTLE CHALLENGES

BY DANIEL CHARYTONOWICZ

ADAPTED FROM CORNELL UNIVERSITY

TODAY YOU HAVE LEARNED A LOT OF NEW THINGS ABOUT YOUR COMPUTER AND PROGRAMMING/CODING. TOGETHER WE WENT THROUGH SOME EXAMPLES IN OUR BLACKJACK GAME, BUT NOW IT IS TIME FOR YOU TO EXPLORE PYTHON ON YOUR OWN. WITH TURTLE YOU HAVE THE ABILITY TO CREATE SOME UNIQUE DRAWINGS. YOU MAY CHOOSE TO EITHER EXPERIMENT ON YOUR OWN, OR FOLLOW THIS SERIES OF 3 "CHALLENGES" WHICH WILL GUIDE YOU ALONG IN TERMS OF WHAT TO DRAW. EACH CHALLENGE IS HARDER THAN ITS PREVIOUS ONE. IF YOU GET STUCK THAT'S OK! WE ARE HERE TO HELP YOU, AND ADDITIONALLY WE HAVE WRITTEN STEP-BY-STEP ANSWERS FOR EACH CHALLENGE THAT YOU CAN FOLLOW ALONG IF YOU LIKE. HAVE FUN AND GOOD LUCK!

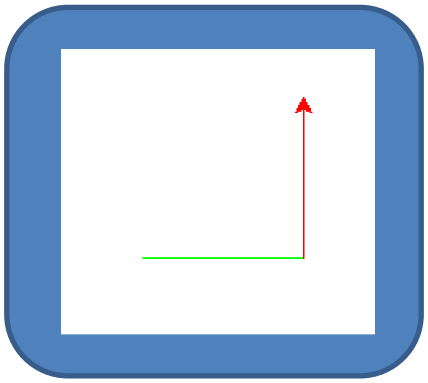
CHALLENGE 1:

IN THE MIDDLE OF THE WINDOW, DRAW A GREEN LINE 100 PIXELS TO THE RIGHT, AND A RED LINE 100 PIXELS UP.

IMORTANT TIPS:

- `turtle.color("Colorname")` changes the color
    - Ex: `turtle.color("Red")` gives red, `turtle.color("Green")` makes it green

Our final product will look like this:



TUTORTIAL:

1. The turtle starts at the center of the window with black colored lines, always facing to the right. So we don't need to do any initial changes in direction. However we do need to change the color of the line because our first line must be green.
2. Our first piece of code will be to change to color of the line:

```
turtle.color("green")
```

3. Now that our line color is green, let's draw our first line. The next piece of code just after changing the color will be:

```
turtle.forward(100)
```

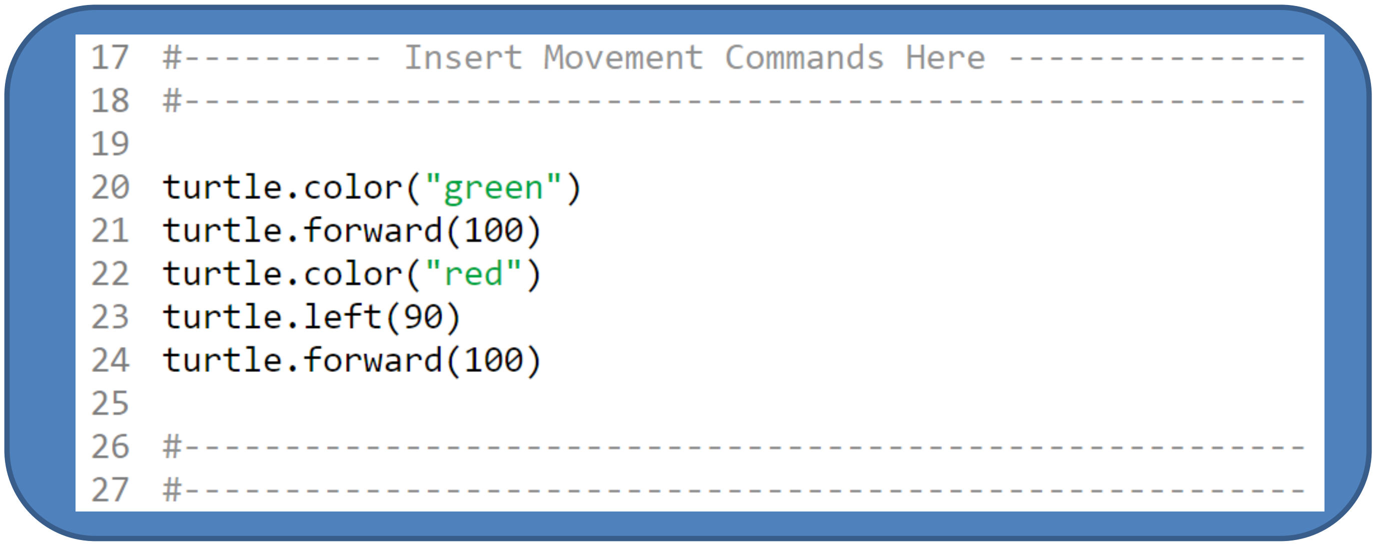
4. You can run the code up to this point and see that we have successfully drawn the green portion of the line. Now we need to do two things: (1) change the line color to red and (2) change the direction our turtle is facing so we can draw up.
5. To solve the two above problems, we will write:

```
turtle.color("red")
turtle.left(90)       #remember turns are done in degrees.
```

6. Now that we have successfully turned our turtle and changed the color, we can go ahead and write the final line of code. This line will be same as the one in step (3).

```
turtle.forward(100)
```

When complete, the entire code should look something like this. Play around with this! Change it! How would you go about making a square shape will different colored sides? A triangle?



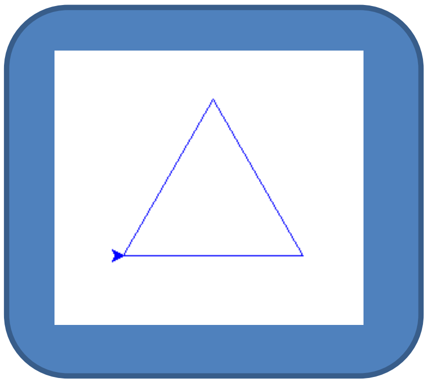
```
#---------- Insert Movement Commands Here ---------------
#--------------------------------------------------------

turtle.color("green")
turtle.forward(100)
turtle.color("red")
turtle.left(90)
turtle.forward(100)

#--------------------------------------------------------
#--------------------------------------------------------
```

CHALLENGE 2:

DRAW A BLUE EQUILATERAL TRIANGLE SUCH THAT EACH SIDE HAS A LENGTH OF 150 PIXELS.

IMORTANT TIPS:

- IN OUR PREVIOUS CHALLENGE, WE MADE RIGHT-ANGLE LEFT TURNS (90 DEGREES). FOR EQUILATERAL TRIANGLES, WHAT INTERNAL ANGLES NEED TO BE MADE? HOW MANY DEGREES DO YOU NEED TO ROTATE TO ACHIEVE THESE ANGLES?



TUTORTIAL:

1. Let's begin by changing the color of our turtle to blue; the entire triangle will be blue, so we only have to do this once. The code for this will be:

   ```
   turtle.color("blue")
   ```

2. Now that our turtle is the correct color we can proceed with drawing the first straight line. The code for this will be:

   ```
   turtle.forward(150)
   ```

3. That does it for our first straight line. Now comes the trickiest part of this challenge: the turn. We know that in an equilateral triangle, the interior angles are each 60 degrees. To create such an interior angle, we need to rotate the turtle 120 degrees left.

   ```
   turtle.rotate(120)
   ```

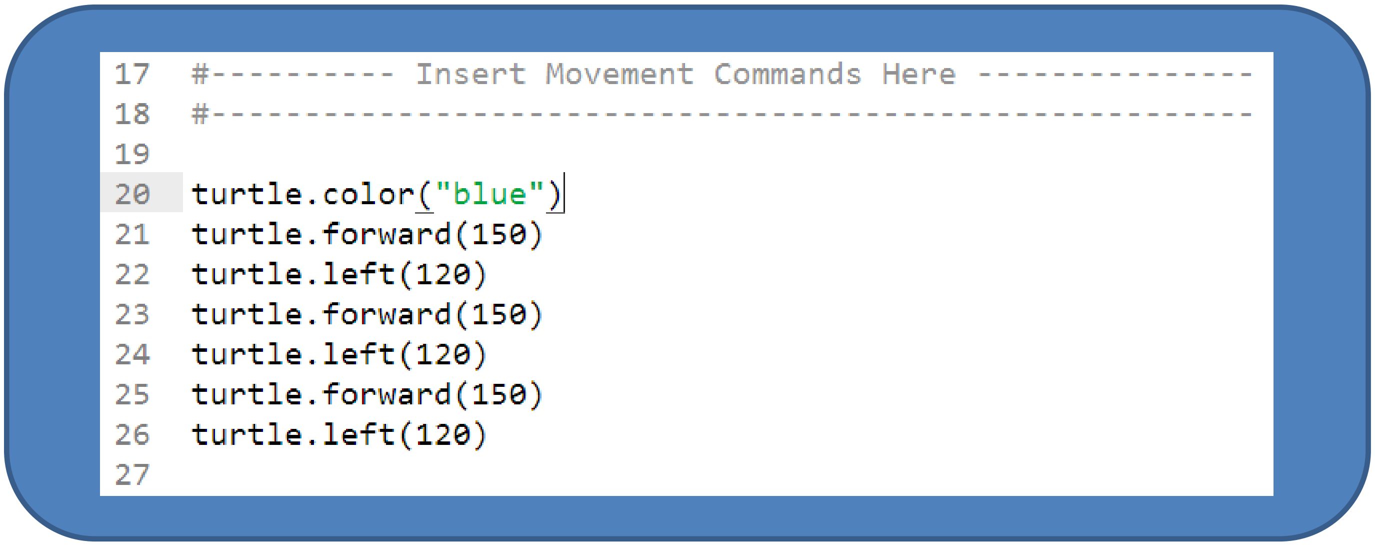
4. We can now go ahead and send the turtle forward another 150 pixels, and again perform a rotation.

   ```
   turtle.forward(150)

   turtle.rotate(120)
   ```

5. The last thing to do here will be to move the turtle forward another 150 pixels.

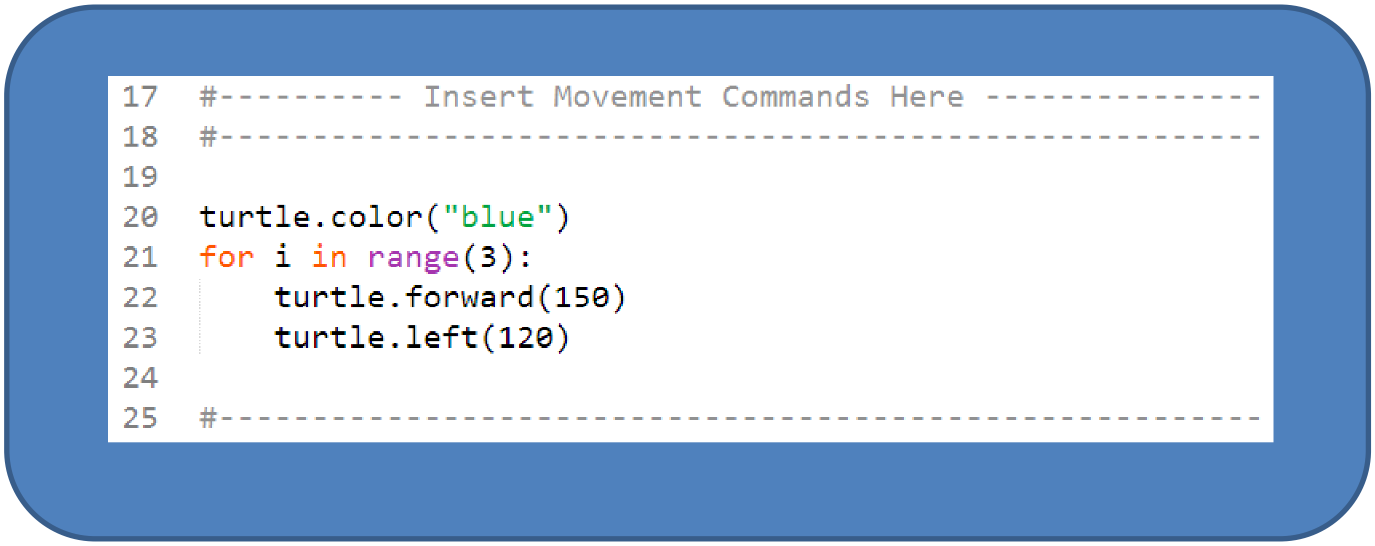
```
turtle.forward(150)
```

6. At this point the triangle should be complete and look similar to the picture above. The total code together will look similar to this:



```
#---------- Insert Movement Commands Here ---------------
#--------------------------------------------------------

turtle.color("blue")
turtle.forward(150)
turtle.left(120)
turtle.forward(150)
turtle.left(120)
turtle.forward(150)
turtle.left(120)

```

There is a simpler way to code this however using for loops. If you think about it, we just made three identical moves forward and rotate commands. Taking this into account, we can rewrite the code to look something like this:



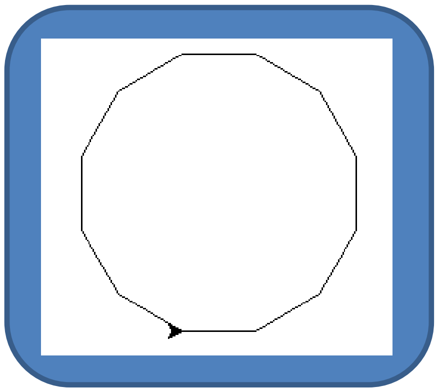
```
#---------- Insert Movement Commands Here ---------------
#--------------------------------------------------------

turtle.color("blue")
for i in range(3):
    turtle.forward(150)
    turtle.left(120)

#--------------------------------------------------------
```

CHALLENGE 3:

DRAW A POLYGON THAT HAS 12 SIDES, EACH 50 UNITS LONG, USING FOR LOOPS.

IMORTANT TIPS:

- REMEMBER THAT ALL POLYGONS (TRIANGLES, RECTANGLES, PENTAGONS, HEXAGONS), WHILE THEY HAVE DIFFERENT INTERIOR ANGLES, ALL INTERIOR ANGLES SUM UP TO 360 DEGREES. TO DETERMINE THE SIZE OF EACH INDIVIDUAL INTERIOR ANGLE (AND THUS HOW MANY DEGREES TO ROTATE EACH TIME) ALL YOU NEED TO DO IS DIVIDE 360 DEGREES BY THE NUMBER OF SIDES.



TUTORTIAL:

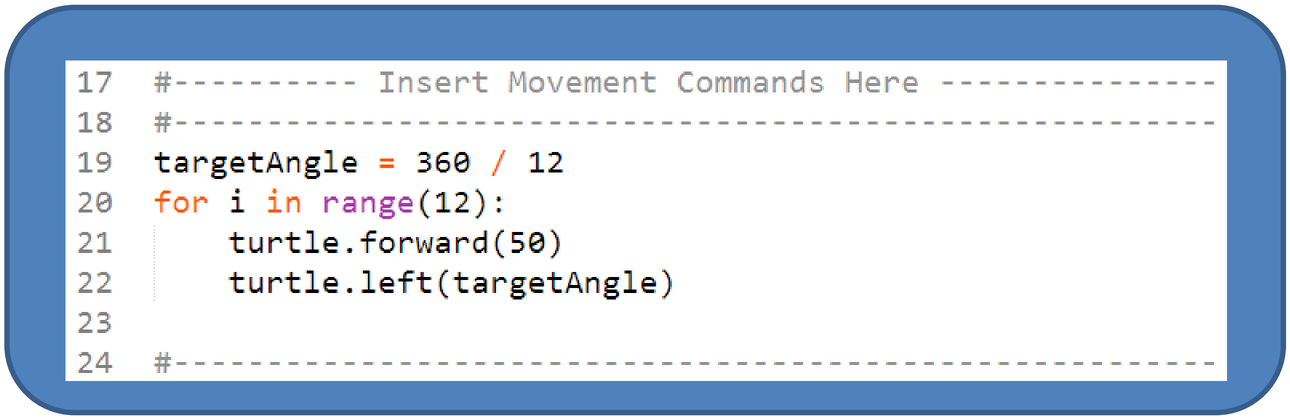
1. Let's begin by determining the target angle we are going to be looking for. To do this, let's create a variable called *targetAngle* and assign it the value of 360/12.

```
targetAngle = 360 / 12
```

2. Now we need to create this polygon. Using what we know from the previous challenge, this polygon can be create using a "for loop" that cycles 12 times. Thus all we need to write is:

```
for i in range(12):

        turtle.forward(50)

        turtle.left(targetAngle)
```

3. That's all! Simpler than you thought right? If you run the code (CRTL+B) you should see the above shape. The final code should look something like this:



```
#---------- Insert Movement Commands Here ---------------
#-------------------------------------------------------
targetAngle = 360 / 12
for i in range(12):
    turtle.forward(50)
    turtle.left(targetAngle)

#-------------------------------------------------------
```