

# **Adrian's CS151 Project 1: What is a Shape?**

# Abstract

For this project, we had to create a series of shapes and patterns by calling feasible commands from Python's pre installed library called turtle. First, I had to create two shapes using Python's turtle by commanding it to create a turtle window in which a called turtle acted like a pen or paintbrush on this turtle window "canvas." These commands include lifting the turtle pen up and placing it down, making it go to a specific spot on the turtle window, making it turn in a certain direction or by a certain number of degrees from its current direction, making it go forward or backward, and making it fill in created shapes. Despite these commands that could result in limitless possible images, I created a blue tipped star and a red pentagon. To make these shapes, I used a loop function in order to repeat steps of drawing the lines that make up the star and pentagon shape instead of having lots of lines of code. Next, we had to create a function that called the previous two functions for shapes in order to create a pattern. I combined the loop functions I used to create my pentagon and star into one function by naming a new function and calling the names of the previous functions within that new function. Finally, I added a scaling parameter and goto command in order to create an image of multiple different sized patterns in different areas of the turtle window. Ultimately, all of the lines of code I created were able to produce the images I expected.

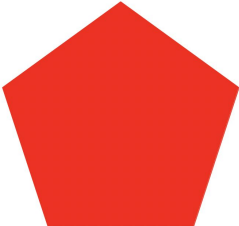
# Results

## First Shape



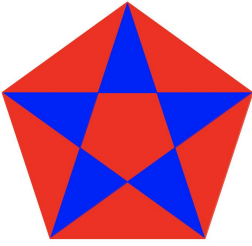
For my first shape, I created a blue tipped star using a turtle with Python in Visual Studio Code with the turtle beginning to draw in the center of the screen. The main component of my code was a loop function that, when later called, repeated a series of starting to fill with a blue color, moving forward (any stated distance), turning left ( $216^\circ$ ), and ending fill five times.

## Second Shape



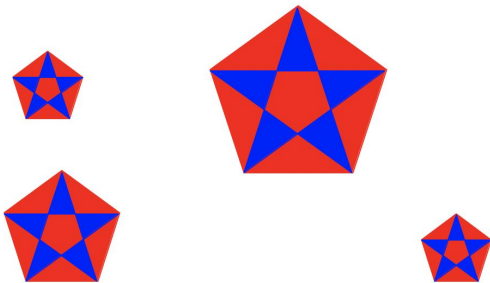
For my second shape, I created a red pentagon using a turtle with Python in Visual Studio Code with the turtle beginning to draw in the center of the screen. The main component of my code was a loop function that, when later called, repeated a series of starting to fill with a red color, moving forward (any stated distance), turning left ( $72^\circ$ ), and ending fill five times.

## First Pattern



For my first pattern, I created a logo consisting of the two previously created shapes (blue tipped star and red pentagon) using a turtle with Python in Visual Studio Code with the turtle beginning to draw in the center of the screen. First, I copied the two functions that when called instructed the turtle to create the two previously created shapes. Then I created a third function that composed these two functions into one that created the logo. In order to draw the star within the pentagon, I had to add lift and drop pen commands and use the go to function to move the turtle to the left upper tip of the pentagon before drawing the star without creating an extra unneeded line on the canvas.

## Second Pattern



Next, I created a pattern consisting of four logos (like the ones created for pattern one) in random locations with different sizes. First, I copied the whole code created for my first pattern. Then, I debugged my code and found that the go to function between making my pentagon and star was unnecessary and actually inhibited me from moving my whole logo on the turtle window screen without dislocating the star from within the pentagon. Finally, I added a scaling command to help draw three logos within the same screen.

# Reflection

During this project, I learned about the seemingly limitless possibilities for how Python turtles can be used. For example, turtles can fill in objects fully or in certain patterns based on how different functions are coded and where begin and end fill commands are placed. In addition, any composite of angles and forward commands can be used, which makes me start to understand how an AI computer generated abstract painting is possible. I also learned about creating a clean organized code with multiple functions, which helped me debug and identify problems that impeded me from obtaining certain images even though previous functions that worked were used.

# Acknowledgements

- 1) I spoke with Jack Nguyen, who took CS151 in the fall, to help debug the final code that created an image of multiple shapes so that each “logo” moved as one shape while also scaling as one shape.
- 2) I consulted lecture notes to help remember how to organize loop functions.