

# Doodle

Read the instructions carefully. Not following the instructions will result in you not getting the credit you want for the assignment.

## Learning Outcomes

1. Use loops, conditional behavior
2. write and call user-defined functions with parameters and return values in your code
3. Avoid writing code with global variables and global code
4. Take program input from the command line terminal, not keyboard

## Problem To Solve

Draw interesting visual shapes, a pattern or a scene on the screen. What you produce does not have to be complex nor "gnarly", just artistic and structured. Likewise, the code is not expected to be complex nor "gnarly", just demonstrate what you've learned--and have fun.

1. **Drawing Shapes:** There are several ways to draw shapes. One way is to draw lines or curves like a spirograph. Another way is to use regular polygons as building blocks for other shapes. For example, a house can be drawn using squares and triangles. Such a house function would call the square and triangle functions you already made. You choose what you want to do.
2. **Scaling Factors:** You will use scaling factors to draw shapes of different sizes. This allows you to create a small house or a large house using the same code. You can scale values in two ways: 1) you can scale the values that get passed to a function that draws some shape, or 2) you can pass a scaling parameter to the function and scale the code inside the function. You choose.
3. **Input:** Your program should take one integer command-line parameter in range  $\{1, 2, 3\}$  that changes some aspect of the drawing for each different value. What that is is up to you. The second parameter should be an output file name. For your submission, call the output file `art.png`, but do not hard-code the file name.

Remember, you may use any code provided in your book as well as using Copilot. Also, don't hesitate to ask for clarification if you're unsure about some resource.

Implement your program in a file called `doodle.py` in a directory called `doodle`.

If your program does not produce a `png` file, there are command line tools like ImageMagick that you can use to convert to `png` format for submission.

## How to Test Your Code

Here are some ways to test your code.

- Test the first parameter with values in the range `{1, 2, 3}` and outside that range.
- Test the first parameter with values that are strings not numbers (bad).
- Test with different output file names.
- Test with no command line input.

## Correctness

GitHub will run some basic checks when you submit your code. Note that many aspects of correctness must be inspected manually versus automatically because the output is visual, so the score may be adjusted after manual inspection of the code and the image your code produces.

## Style

Execute the below to evaluate the style of your code.

```
ruff format doodle.py  
ruff check doodle.py
```

## How to Submit

From your assignment repository page in GitHub Classroom, click "Submit" button and enter a commit message.