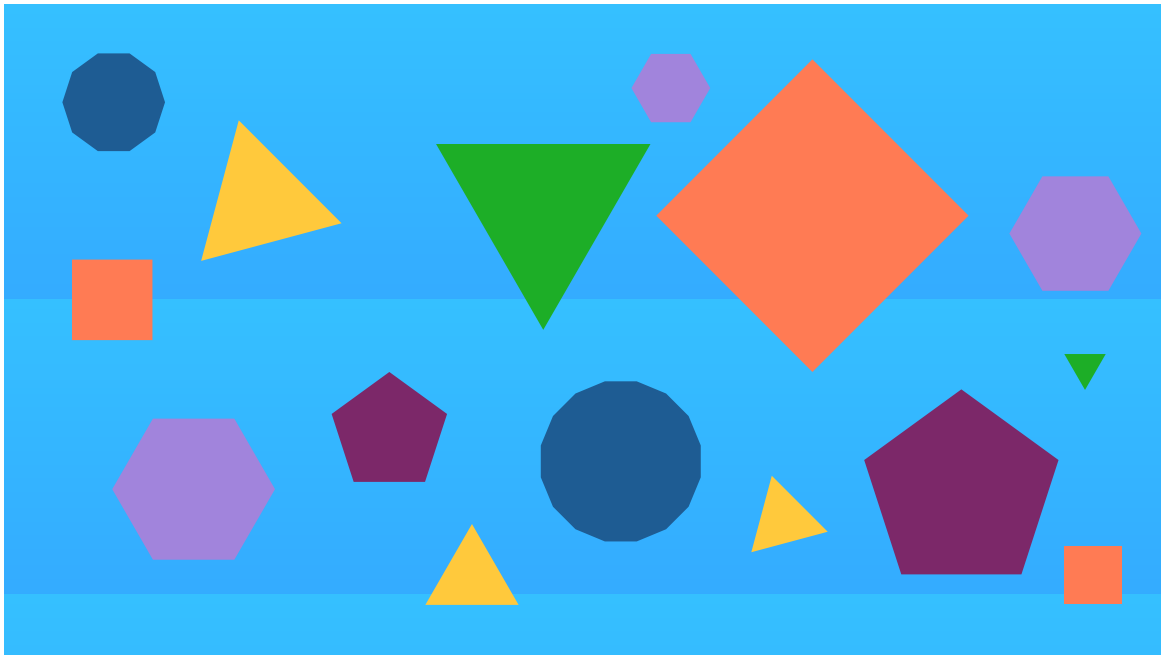




Geometric Art: Drawing Polygons

Explore the power of 'While' loops to programmatically draw perfect geometric shapes, from triangles to hectogons.

Courses

- Grades 6-12

Materials

- Cell phone, tablet, or computer
- Internet connection

Educational Objectives

- Understand the concept of a "while" loop in programming.
- Create a technological object (prototype) using a device.
- Identify relationships between technology and the surrounding world.
- Evaluate personal and peer work.
- Engage in dialogue and reflection on ideas for improvement.

Start (10 minutes) - The Conditional Loop

1. Welcome students and introduce the day's activity: **"Today we will learn to draw complex geometric shapes using code."**
2. Introduce a new kind of loop. Ask the class: **"We've seen a loop that runs forever. But what if we want a loop to run only *while* a certain condition is true, and then stop?"**
3. Explain that this is what a "while" loop does. Then, connect it to a real task: drawing a polygon. Ask, **"How would we draw a square? We draw a side, turn 90 degrees, draw another side... we repeat the 'draw and turn' action *while* we still have sides left to draw."** This is the perfect introduction to the logic they'll be building.

The 'Repeat While' Loop

A **'repeat while'** loop is like a persistent guard. It first checks a condition. **While** that condition is true, it runs a block of code over and over again. The moment the condition becomes false, the loop stops. This is perfect for

situations where you don't know the exact number of repetitions in advance, but you know the goal you're trying to reach.

What is a regular polygon?

A regular polygon is a shape where all sides have the same length and all interior angles are equal. You know the simple ones: an equilateral triangle (3 sides), a square (4 sides), a pentagon (5 sides). But what about a decagon (10 sides) or an icosagon (20 sides)? Drawing these by hand would be very tedious!

The Geometry Secret: 360 Degrees

Here's a cool geometry trick: the sum of the exterior angles of *any* regular polygon is always **360°**. So, to figure out how much our "pen" needs to turn at each corner, we just divide 360 by the number of sides! - **Triangle**: $360 / 3 = 120^\circ$ turn - **Square**: $360 / 4 = 90^\circ$ turn - **Pentagon**: $360 / 5 = 72^\circ$ turn

Let's combine loops and geometry!

Notice the repeating pattern for drawing any polygon: **draw a line**, then **rotate**. We repeat this process until we've drawn all the sides. This is a perfect job for a `repeat while` loop! We can tell the program: **WHILE** the number of sides we've drawn is less than the total number of sides we want, **DO** the "draw and rotate" action.

Development (20-30 minutes) - The Polygon Generator

1. Now that the students understand the logic of using a 'while' loop and the simple math behind polygons, it's time to build their shape generator.
2. Lead them through **the instructions for creating the program that draws polygons based on the knob's input**, as detailed in the hands-on section below. Pay close attention to the `while` loop's condition (e.g., `while counter < sides`) and how the turning angle is calculated.

Closure (5-10 minutes) - Creative Coding

1. Once everyone has an interactive polygon generator, it's time to experiment and play the role of a creative coder.
2. Use the final section to challenge them to modify the program's variables. This encourages them to see how small changes in the code can lead to big changes in the visual output, which is a core concept in generative art.

Challenge: Creative Coding Mods

Now that you've implemented the polygon generator, let's play with the code!

- **Mod #1: Perimeter Control.** How would you change the code to make the polygon bigger or smaller?
- **Mod #2: Spin Direction.** What happens if you rotate to the right instead of rotating to the left?
- **Mod #3: Dynamic Color.** How can we change the color of the lines based on the number of sides of the polygon?