



Practice 9: Polygon Generator

Creating functions (commands) with parameters and abstraction
Module 3: Functions and Abstraction

The magic of creating your own blocks

Until now, we have used blocks that were already created in **Snap!**. But what if we could create our own blocks? **Abstraction** consists of grouping a lot of complicated steps under a simple name. Today we are going to teach the computer to “draw polygons” with just one word.

Practice Objectives

- Learn to create custom blocks of type **command**.
- Understand the concept of **Parameter** (data we send to the function).
- Apply geometry to draw any regular polygon.
- Improve code readability and efficiency.

Key Concept: The Parameter

A **parameter** is like a “variable” that lives inside the block.

- **Why use them?** If we create a block `draw_square`, we can add a parameter called `side`.
- **Versatility:** This way we can decide if the square is small (50 steps) or large (200) without having to write the code twice.

THE CHALLENGE: The Geometric Garden

Create a program where the sprite draws several figures using your own blocks.

Suggested steps:

1. Create the "Square" block:

- In the block palette, click on +. Category **Motion**.
- Add an *input* (parameter) named *size*.
- Logic: Repeat (4) [Move (size), Turn (90) degrees].

2. Create the "Universal Polygon" block:

- You will need two parameters: *sides* and *length*.
- **Formula:** The turn angle is always $360 / \text{sides}$.

3. Main program: When the green flag is clicked, the sprite should:

- Draw a blue **triangle**.
- Move and draw a red **square**.
- Move and draw a green **hexagon**.

Code Example (JavaScript)

In professional programming, functions with parameters are the base of everything. See how what you just did in Snap! would be written using text:

```
// Function to draw any polygon
function drawPolygon(sides, length) {
  let angle = 360 / sides;
  for (let i = 0; i < sides; i++) {
    move(length);
    turn(angle);
  }
}

// Calling the function
drawPolygon(5, 100); // Draws a pentagon
```

Important Difference

- **Without functions:** If you want to draw 10 squares, you have to repeat the "repeat 4" block ten times. Your code becomes long and hard to read.
- **With functions (Abstraction):** Your main program only says draw square. It is much cleaner and more professional.

This document is published under license
© Creative Commons Attribution 4.0 International (CC BY 4.0)