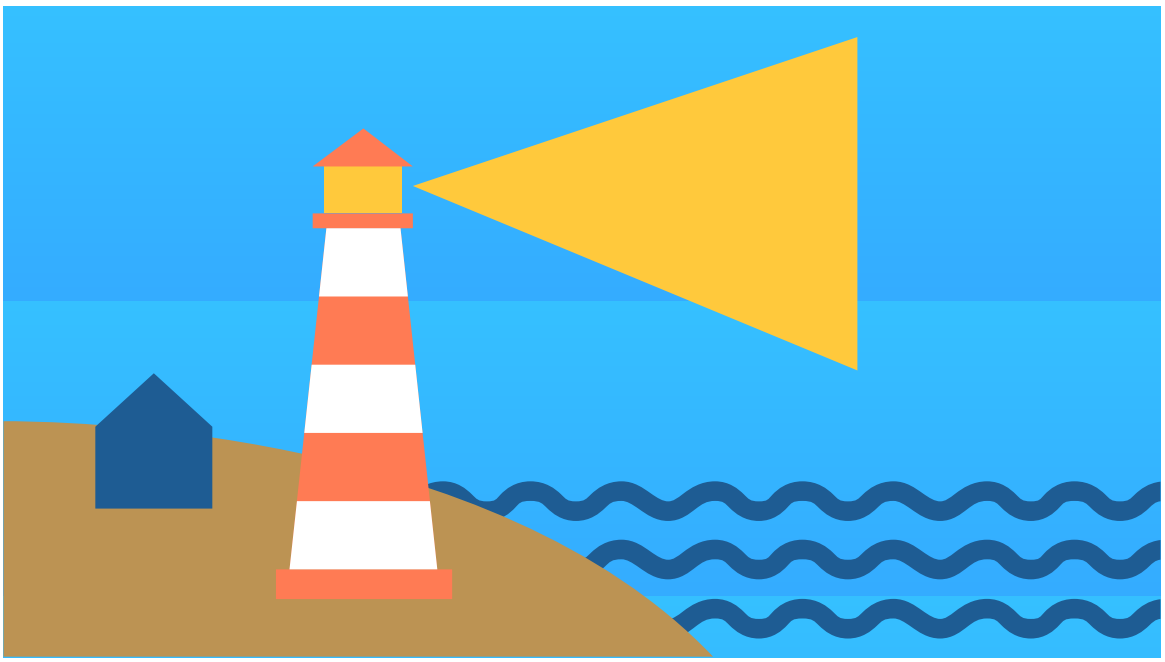

The Lighthouse

Master the 'For' loop to create a beautiful, gradual flashing effect for a lighthouse that you can turn on and off.

Courses

- Grades 6-12

Materials

- Cell phone, tablet, or computer
- Internet connection

Educational Objectives

- Understand the concept of a FOR loop.
- 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 improvement ideas.

Start (10 minutes) - The Controlled Loop

1. Welcome students and introduce the day's activity: **"Today we will learn to prototype a lighthouse with a gradual, glowing light."**
2. Introduce the new programming tool for today. Ask the class: **"What if we need a loop that runs an exact number of times, like 100 steps to go from dim to bright?"**
3. Explain that while a 'forever' loop is great for endless automation, we sometimes need more control. This is where the **'For' loop** comes in. Break down its components for them: a counter variable, a start value, an end value, and a step size. This is the perfect tool for creating smooth, gradual animations.

The FOR Loop

A **'count with variable'** loop, known in most programming languages as a **'for'** loop, is designed to repeat a block of code a specific, known number of times. It uses a counter variable to keep track of its progress from a start point to an end point.

Components of the FOR loop

Think of a **'For' loop** like a mission plan: 1. **Variable**: Your counter or "agent" that will be doing the counting (e.g., `i` or `intensity`). 2. **Initial value**: The starting number for your counter. 3. **Final value**: The target number where the loop will stop. 4. **Steps**: How much the counter changes in each repetition (e.g., counting up by 1, or down by 1).

How do we use it for a gradual glow?

A **'for'** loop is perfect for animation because it runs for a known number of steps. To make our lighthouse light up gradually, we'll use a 'for' loop that counts an `intensity` variable from 0 to 100. In each step, we set the light's brightness to the current value of `intensity`. This creates a smooth fade-in. To make it dim, we'll use a second 'for' loop that counts `intensity` from 100 back down to 0!

Development (20-30 minutes) - Building the Glow

1. Now that the students understand the mechanics of a 'For' loop, it's time to build the gradual glowing effect.
2. Lead them through **the instructions for creating the lighthouse prototype**, as detailed in the hands-on section below. Pay close attention to how they will use two 'For' loops: one to count up (fade in) and a second to count down (fade out).

Closure (5-10 minutes) - The Great Loop Debate

1. Once everyone has a working lighthouse with a smooth pulse, it's time for a deeper dive into loop theory. This is a key conceptual moment.
2. Use the final section to guide a critical discussion comparing the 'For' loop with the 'While' loop they've seen before. Challenge them to rebuild this project with the "wrong" tool to see why programmers choose different loops for different jobs.

Reflect: FOR vs. WHILE

You've now used different kinds of loops. Let's compare them.

- What is the main difference between a **'for' loop** (count with variable) and a **'while' loop** (repeat while)?
- A 'for' loop is great when you know the exact start and end points (like 0 to 100). A 'while' loop is great when you just need to wait for a condition to change.
- Are there cases where you could use either one? Absolutely!