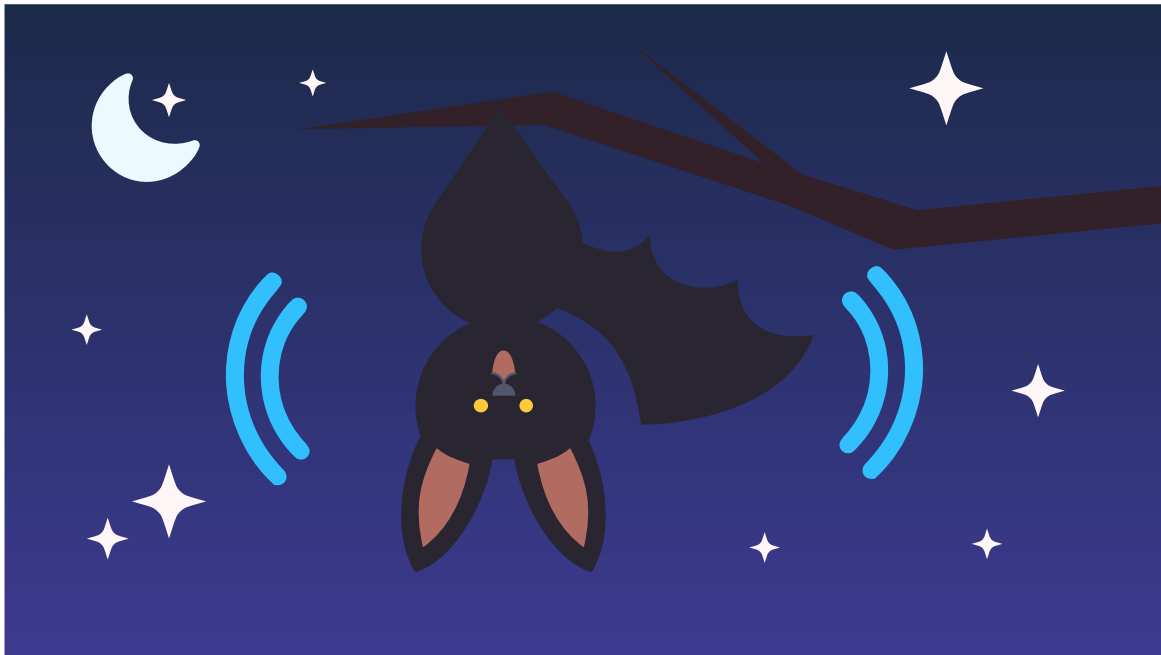




Flight of the Night Bat

Strengthen your command of 'If... Else...' logic by flipping the condition around. Create a virtual bat that only wakes up and takes flight when the room goes dark.

Courses

- Grades 6-12

Materials

- Mobile phone, tablet, or computer
- Internet connection
- A light source and a place where you can make it dark.
- Cardboard, scissors and a couple of rubber bands to hold the phone against the cardboard

Description

This activity is designed specifically to reinforce the "IF-ELSE" concept with an inverted condition (using the "less than" operator instead of "greater than"). The bat is the perfect metaphor for understanding why a program would wait for darkness before springing into action.

Educational Objectives

- Consolidate their understanding of the "IF... ELSE..." conditional structure.
- Practice using different logical comparison operators (in this case, "<").
- Show how a small change in the logic can completely reverse a program's behavior.
- Apply conditional thinking to simulate the behavior of a living creature.

Start (10 minutes) - Inverted Logic

1. Welcome the class: **"Today we're going to show just how powerful and flexible conditionals are. We'll take a logic you already know and turn it upside down to create an animal behavior."**
2. Ask the class: **"Many programs spring into action when there's light. But which living creatures come alive specifically in the dark?"** (Main example: the bat).

3. Explain the plan: **"We're going to build a bat that wakes up when it gets dark. We'll use the same 'IF... ELSE...' structure we already know, but we'll change the 'question' we ask the light sensor so that it detects the absence of brightness."**

Reinforcing the 'If... Else...'

The best way to master a concept is to practice it. Let's go over the **"If... Else..."** structure again: it's like a decision the bat makes before it acts. 1. The program reaches a **condition** (Is there very little light?). 2. If the answer is **true**, it takes the first path (Time to fly!). 3. If the answer is **false**, it takes the second path (Keep on sleeping). Once you master this structure, you can control the behavior of any digital creature.

Changing the Question: Comparison Operators

For this project, we'll simply switch the operator to **"less than" (<)**. With `light > 50` a program activates in **brightness**; with `light < 30` it activates in **darkness**. Same block, opposite question. Now, instead of springing into action when it's bright, our program will be watching for darkness so it can wake the bat up!

The Bat's Logic

The logic for our bat is a **mirror image**: where other programs look for light, this one looks for its absence. * **Condition:** Is the light sensor value **less than 30**? * **IF TRUE (it's dark):** The bat wakes up! The flying animation runs. * **ELSE (it's FALSE, there's light):** The bat "sleeps", which means it disappears from the screen.

Development (20-30 minutes) - Programming the Bat

1. With the goal of inverting the logic now clear, students can start building.
2. Guide them through **the instructions for creating the bat**, as detailed below. Encourage students to notice that the structure doesn't change: the operator does.

Closing (5-10 minutes) - A Small Change, a Big Impact

1. Once everyone's bats are flying in the dark, it's time to reflect.
2. Kick off the discussion: **"Look at your own code. How much of it would change if you wanted the opposite behavior? What is the one block that decides whether the creature loves the day or the night?"**

Reflect

How do you think this kind of logic helps engineers build streetlights that switch on by themselves?

What if the bat were really shy and you needed it to be *completely* dark before it came out? Would you raise or lower the number 30?

Can you think of any other animal that follows this "If it's dark, then..." logic? (For example: owls, fireflies).