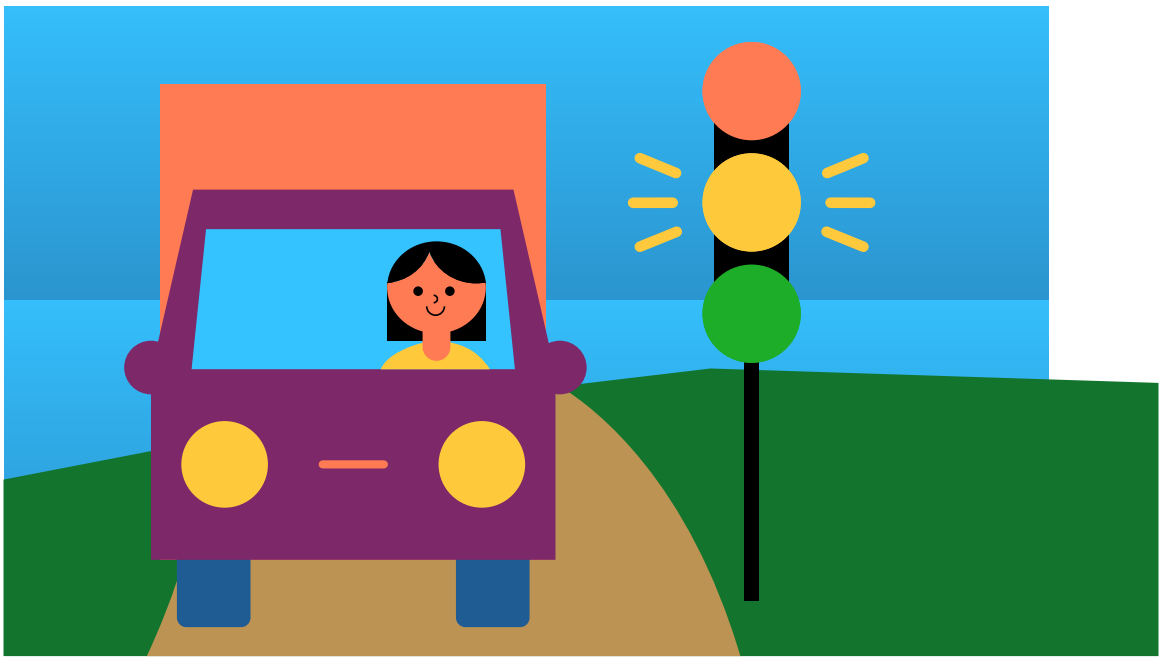

# A Flashing Warning

Dive deeper into automation by nesting loops to create a realistic, flashing yellow traffic light.

## Courses

- Grades 3-12

## Materials

- Cellphone, tablet, or computer
- Internet connection

## Educational Objectives

- Understand the concept of "loop," both general and nested.
- Develop a technological object (prototype) using a device.
- Identify relationships between technology and the surrounding environment.
- Evaluate one's own work and that of others, both individually and in teams.
- Participate in dialogues and reflections to propose improvements.

## Start (10 minutes) - The Flashing Challenge

1. Welcome students and introduce the day’s activity: **"Today, we are going to learn how to create a more advanced and realistic traffic light."**
2. Ask the class to think about real-world traffic lights: **"What does the yellow light often do right before it turns red to get your attention?"** Guide them to the idea of a flashing warning.
3. Pose the programming challenge: **"How would we code that? We could copy-paste the ON/OFF blocks for the yellow light, but our code would get long and messy very fast!"** This introduces the problem of repetition.
4. Explain that to solve this efficiently, we need a new kind of loop—one that can run a specific number of times *inside* our main "forever" loop. This is the perfect introduction to the concept of nested loops.

## How can we program a flashing light?

Imagine we've built an automated traffic light that cycles through its colors inside a "repeat forever" loop. It's a great start, but we can make it more realistic. What if we wanted the **yellow light to flash** a few times before turning red? This would be a much better and safer warning for drivers.

## So, do we just add more blocks?

To make the light flash, we'd need to add a sequence like this to our main loop: **Green → Yellow ON → OFF → Yellow ON → OFF → Yellow ON → OFF → Red** This works, but it's just repeating the **"Yellow ON → OFF"** sequence. Once again, **our code would become very long and repetitive!** There must be a smarter way.

## Why Not Another 'Forever' Loop?

You can't have two captains on one ship, and you can't have more than one "repeat forever" (Main Loop) in a program! If you did, the computer wouldn't know which one is the main program or what to run. It would get confused!

## The Solution: A Loop Inside a Loop!

This is where a different kind of **loop** comes in handy! We can use a loop that repeats a specific number of times (in the image, 3 times). This is perfect for the flashing part. Even better, we can **nest loops**. This means putting one loop *inside* another. Repetitions within repetitions!

## Flashing Yellow: A Nested Loop

By using a numbered loop, we can repeat the **"Yellow ON → OFF"** blocks as many times as we need. By putting this smaller loop *inside* our main "forever" loop, we create a powerful and efficient program: Main Loop ( **Green** → *Inner Loop* ( **Yellow ON** → **OFF** [x4] ) → **Red** )

## Development (20-30 minutes) - Building with Nested Loops

1. Now that they've grasped the concept of nesting one loop inside another, it's time to build the advanced traffic light.
2. Lead them through **the instructions for modifying the traffic light sequence**, showing them exactly where to add the new numbered loop to create the flashing yellow effect. This will be their first time creating a loop within a loop.

## Closure (5-10 minutes) - Seeing Loops Everywhere

1. Once everyone has a working, flashing traffic light, broaden the discussion to think about where else this powerful concept applies.
2. Use the final section to spark a creative discussion about loops in everyday life and challenge them with an even more complex traffic simulation.

### Reflect

---

**You've now mastered nested loops! Ask yourself:**

- Loops aren't just for code; they're everywhere. Where can you see repeating patterns or cycles in everyday life (think music, daily routines, nature)?
- What other cool prototype could you build using a loop inside a loop?