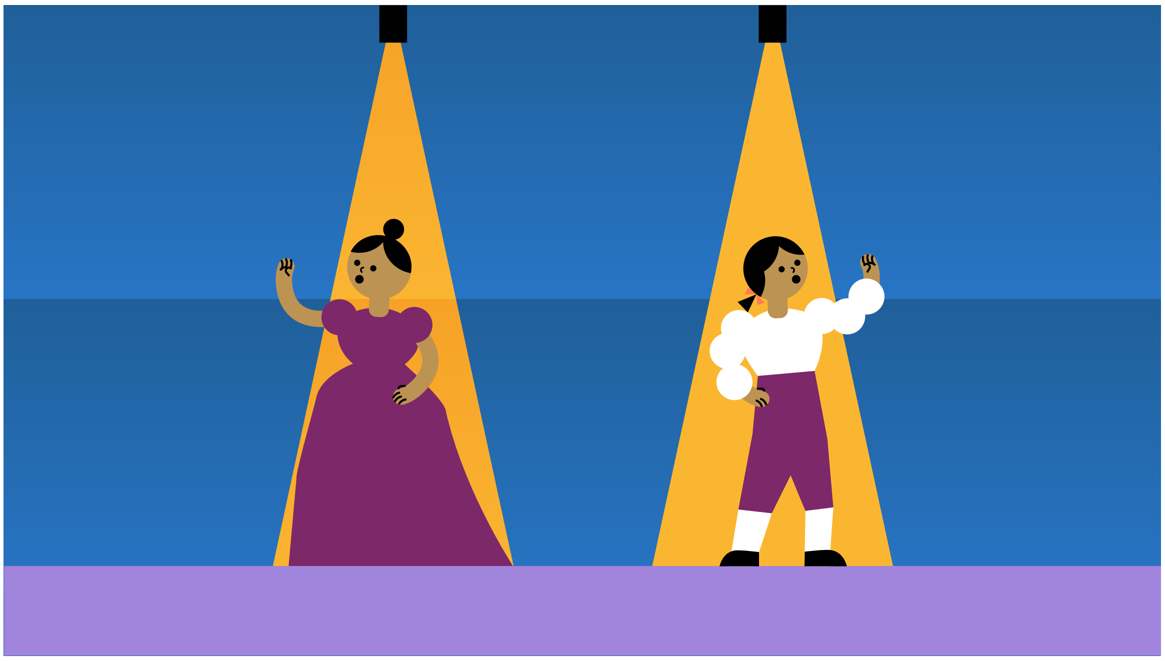

Smart Code: Using Functions

Learn to write cleaner, reusable code by creating functions with parameters for a two-button light game.

Courses

- Grades 6-12

Materials

- Cell phone, tablet, or computer
- Internet connection

Educational Objectives

- Understand the concept of functions and parameters.
- Create a technological object (prototype) using a device.
- Identify relationships between technology and the surrounding world.
- Evaluate personal and others’ work.
- Engage in dialogue and reflection on improvement ideas.

Start (10 minutes) - The Problem with Repetition

1. Welcome students and introduce the day’s activity: **"Today we will learn how to write smarter, more efficient code using functions."**
2. Kick off the lesson by asking: **"Imagine you're writing a program and you need to use the exact same 10 blocks of code in five different places. What's the problem with just copying and pasting them?"** (Guide them to issues of size, messiness, and the difficulty of making changes—you'd have to fix a bug in 5 places!).
3. Explain that programmers have a powerful solution for this: **Functions**. Use the analogy of a recipe: instead of writing out the steps every time, you just refer to the recipe name. This is the perfect entry point to explain how functions help us write reusable code.

What are functions?

Functions are reusable groups of code that you can name and then "call" whenever you need them. Imagine you have a favorite recipe with five steps to make dinner. Instead of writing those five steps out every single time, you can create a single "recipe card" named "DINNER". Now, you only need to call that one function, and the program runs all five steps. This is a function **without parameters** because it does the exact same thing every time.

Functions with Parameters

But what if you want to make the same recipe with different main ingredients? We can make our function smarter by giving it **parameters**—inputs that can change the result. Think of the DINNER function again. We can add a parameter for the main ingredient. Now, calling `DINNER(noodles)` will give a different result than `DINNER(rice)`. The steps are the same, but the input changes the outcome. In our activity, we will create a `light_sequence` function with a **color** parameter. The function will always perform the same blinking pattern, but the color of the light will change based on the parameter we provide when we call it.

Development (20-30 minutes) - Building a Reusable Function

1. Now that the students understand the power of creating reusable code blocks, it's time to build their first function.
2. Lead them through **the instructions for creating their own `light_sequence` function and then calling it from two different button events**, as detailed in the hands-on section below. This is their first time defining a custom block and then using it in their program, which is a major step.

Closure (5-10 minutes) - The Power of Parameters

1. Once everyone has their two-button light panel working from a single function, it's time to reflect on the power and elegance of this new tool.
2. Use the final section to encourage experimentation with the function's parameters and to challenge them to build a second, different function for another task.

Reflect

You've created your first function! Now it's time to experiment.

- How would you add a third button that calls the same function but with the color **blue**?
- How would you change the speed of the blinking? (Hint: you only have to edit the function in one place!)
- What if you wanted to create a second, completely different light pattern? Would you create a new function?