

Lesson 2: Multi- Screen Apps



Adapted from code.org curriculum

Objectives: You will be able too...



- ✧ Write a simple event-driven program that has multiple screens.
- ✧ Recognize debugging as an important step in developing a program.
- ✧ Use `console.log` to debug simple issues related to event-driven programming.

Getting Started



-
- ☞ In the last lesson you ended up making a simple “chaser game” that wasn’t much of a game.
 - ☞ In this lesson you’ll learn improve that app by:
 - ☞ adding more screens
 - ☞ and adding a way for the game to end.

Activity: Code Studio



- ⌘ Today we will still work independently
- ⌘ You can talk through ideas with your partner
- ⌘ You can read instructions together, and ask questions of each other.
- ⌘ In particular, it's effective to do prediction tasks with a partner.

Programming Language:



- ❧ Every programming language has some way of displaying simple plaintext output
- ❧ Printing to the console is a very common technique not only for debugging but also as an aid in program construction and development.
- ❧ In other words, you can use `console.log` to send messages to yourself to verify the program is doing what you think it's doing which helps prevent errors down the line.

Wrap-up:



- ☞ Today was one of the first times we saw error messages in our programs and started thinking about debugging our code.
- ☞ Is it “bad” to generate an error message? Will every error in our programs generate an error? Why might a programmer actually “like” to get an error message?

Wrap up:



- ❧ Even expert programmers make errors, so debugging is a critical step of writing any program.
- ❧ Since we can assume that all code will have some errors in it, we'd much prefer the computer to catch those errors for us. Error messages are how the computer gives you a helping hand in writing your program, and often they'll include helpful information about how you can fix your code.

Wrap up:



- Of course, not every error will generate an error message because sometimes we write functional code that does something different than we want. In order to catch these logical errors, we'll need to understand how our code is supposed to run and then test it to make sure that it does.
- In either case, this process of finding and fixing errors in your code is entirely normal and is just as important a skill as writing the code in the first place.

Wrap-up



- ❧ We're making a big deal out of error messages and debugging because they are often hurdles for new learners.
- ❧ But you just need to have the right attitude about writing code - debugging is part of the process.
- ❧ You get used to a pattern of:
 - ❧ Write a little code
 - ❧ Test it to make sure it does what you think
 - ❧ Write the next piece
- ❧ If you do this, the errors you make will tend to be smaller and easier to catch.

Vocabulary:



- ⌘ Debugging - Finding and fixing problems in your algorithm or program.
- ⌘ Event-driven program - a program designed to run blocks of code or functions in response to specified events (e.g. a mouse click)
- ⌘ Event handling - an overarching term for the coding tasks involved in making a program respond to events by triggering functions.