Lesson 7: If-statements unplugged

Adapted from Code.org curriculum

Objectives: You will be able too…

- Reason about if-statements by tracing pseudocode programs by hand
- Write a short program in pseudocode that uses if statements
- Explain the purpose of if-statements in programs

Vocabulary:

- Conditionals - Statements that only run under certain conditions.
- If-Statement - The common programming structure that implements "conditional statements".
- Selection - A generic term for a type of programming statement (usually an if-statement) that uses a Boolean condition to determine, or select, whether or not to run a certain block of statements.

When vs. If

- Most of the programs you've written so far have event handlers that get triggered when certain events occur.
- But in the last program - the version of "Apple Grab" - we had a very simple if statement that said something like:
  - if(count==20){
    - setScreen("gameOver");
  - }

When vs. If

- The introduction of “if” introduces an English language issue for us moving forward. Here is an example:
- I’m going to read out loud two sentences that describe a program. With a partner discuss what the difference is between them, and decide which one is “right”. Here are the two sentences:
  - When the button is clicked add one to the score.
  - If the button is clicked add one to the score.

When vs. If

- Let’s try another one:
  - When the score reaches 20, set the screen to “game over.”
  - If the score reaches 20, set the screen to “game over.”

When vs. If

- There is no right answer. In English both pairs of sentences mean basically the same thing.
- However in programming, using the words “if” and “when” map to some expectations about how the underlying code is written.

When vs. If

- Here is the difference:
  - **"When"** is used in reference to an **event –** When something happens respond in such and such a way.
  - **"If"** is used in reference to a **decision** about whether or not to execute a certain piece of code – If something is true, then do this, otherwise do that. (conditional logic)
- When describing the behavior of a program events and decisions might get mixed together. For example:
  - "When the button is clicked, if the score is 20 go to 'game over', otherwise add one to the score".

Today…

- Today’s activity focuses solely on if statements
- If the distinction between “when” and “if” is still a little fuzzy, that’s okay
- For now, the key idea is that if statements are a new entity that let us do things we could not do with event handlers – writing code to make decisions about whether or not to run some other piece of code

Code Studio

- Big Picture – Let’s talk about Pseudocode

“Will it Crash?”

- We will use the “Will it Crash? – Activity Guide”
- Homework: Finish “writing” the code

Wrap-up:

- Were you tripped up by any of the problems? Which ones? Why?
- What's the difference between a seqeunce or series of if statements versus an if-else statement?

Wrap-up:

- If-statements and conditional expressions are huge part of programming and we’re going to spend some time digging in with them.
- There are two main issues to concern yourself with when it comes to if-statements and today we’ve looked a lot at one of them, namely, **program flow and order of execution.**
- For example, one very common misconception, or place where people get tripped up is, in the **difference between a sequence of if-statements, and using an if-else statement.**

Creativity in Programming

- How many different coding solutions to the last problem were there?
- Why are different solutions possible?

Creativity in Programming

- There are multiple correct solutions
- This is because there are multiple ways to think about the problem
- There are also multiple algorithms for solving it
- Even if you used the same algorithm, the code might be different.
- All of this demonstrates that **programming is a creative activity**.