Logic, Loops, & Conditionals

If you have any corrections or suggestions to make this write up better, please let us know HERE. We want to hear from you!

For more information on digital literacy, or for additional resources, please refer to the session presentation HERE.

Science	Practical & Applied Arts
Grade 1 Science – SE1.1 Grade 5 Science – HB5.5	ROBA26 ROBA28A ROBA65 ROBA66A

Arts Education	Mathematics	Physical Education
Grade 1 Art - CP1.1 Grade 2 Art - CR2.2 Grade 3 Art - CP3.1	Grade 1 Math - P1.1 Grade 2 Math - P2.1 Grade 3 Math - P3.1 Grade 4 Math - P4.2	Grade 1 - PE1.4 PE1.9 Grade 2 - PE2.10 Grade 3 - PE3.8 Grade 4 - PE4.12

Activity in this Package!

Activity 2: Coding Maze Mat-ness

Additional Resources

Inspiring learning through play! https://www.thinkfun.com/teachers/downloadable-games-brainteasers/

Unplugged Coding Websites:

- https://www.csunplugged.org/en/
- https://hourofcode.com/
- https://www.madewithcode.com/projects/

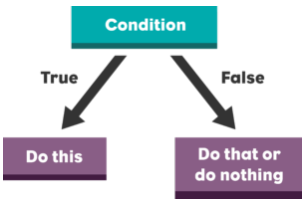
Looking for more ways to use the Let's Go Code! Activity Box? Take a look here for more cool ideas!
Youtube videos:

- If you are needing a little help understanding how to use the Let's Go Code! Activity Box click here

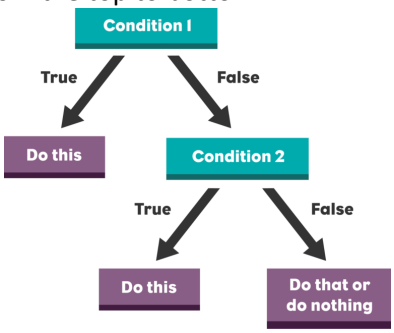
Big Ideas

Conditional Statements

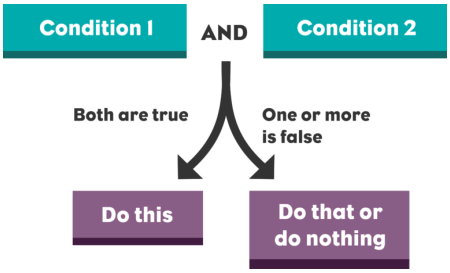
- **If/Else**- the "if" statement tells the computer what to do if the condition is true. The "else" statement tells the computer what to do if the condition is false
    - ex. *If* the number is divisible by 2 it is even, *else* it is odd.



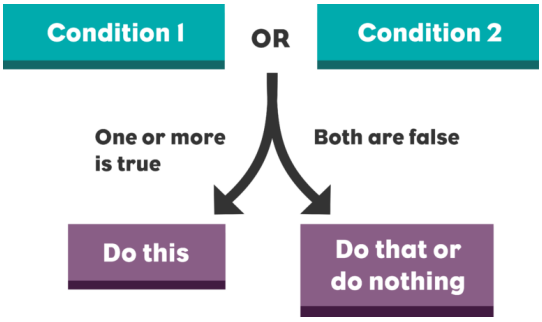
    - Multiple conditions can be added by inputting an additional else if statement between the initial conditions
    - ex. *If* you are free tonight we can meet then, *else if* we can meet tomorrow *else* we can meet on Saturday.
    - conditions are tested from the top to bottom



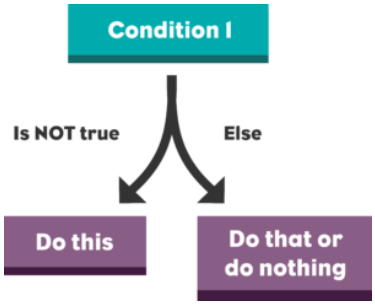
- **Logical Operators:** combine many boolean statements together (Checkpoint: **Boolean statements** are statements that are either true or false)
    - three main operators: and, or, & not.
    - **And:** for an "and" statement to be true, all of its' conditions must be true
        - True and True = True
        - False and True = False
        - True and False = False
        - False and False = False



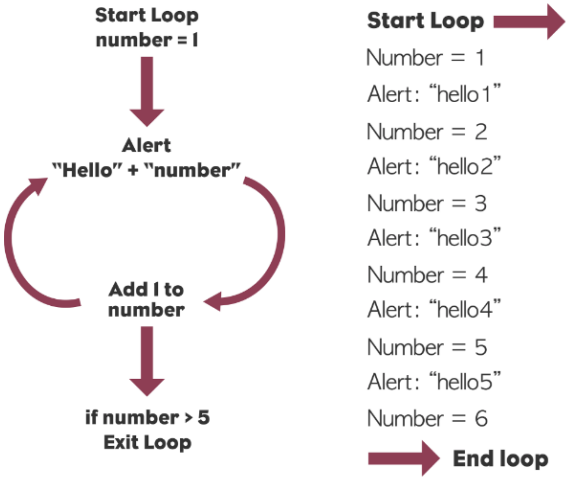
- **Or:** for an "or" statement to be true, at least one of its' conditions must be true
  - true or true = true
  - true or false = true
  - false or true = true
  - false or false = false



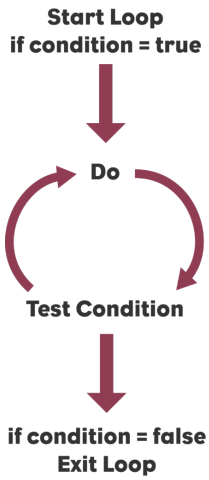
- **Not:** switches the value of the boolean to be the opposite of what it is
  - Not true = false
  - Not false = true



- **For Loops**-repeat a block of code a specified number of times. You determine how long your loops will go *for*
  - often use a variable, defined as counter, inside your loop to count how many times your loop has been run



- **While Loops:** loops that will continue to go until the condition(s) are no longer true
    - ex. **while** (amount of people at a party) > 0, **do** play music.
    - As demonstrated above there are two parts to a while statement: while and do



Quick Reference Terminology

Review of key words:

- Booleans - datatype that can be true or false
- Conditions - something an app evaluates to be true or false
- Conditional Statement - tells the app what to do after evaluating conditions
- AND Operator - evaluates as true if all of the inputs are true

- OR Operator - evaluates as true if one of the inputs is true
- NOT Operator - evaluates as the opposite of the input
- Loop - a repeating block of code
- For Loop - repeats a block of code a certain number of items
- For Each Loop - repeats a block of code for the number of items in a list
- While Loop - repeats a block of code while a condition is true

Real-world Applications: The other meaning of 'Logic'

...afterall, why is teaching logic important??

What is logic?

Understanding the world around us on a deeper level is something we want all of our students to achieve. Whether they are just starting their academic careers, in grade one, or reaching the bring of teenage-dom, all students should be taught the importance of logic.

Logic is the part of science that informs our students about validity and falsity in varying situations. It is the role of logic that helps students to decide what is true or false based on principle facts and inference.

Logic is separate from reasoning because logic forces concrete examples to support or deny the claims presented by specific situations. Our students will experience a myriad of different experiences in their adolescent years. We need to be teaching them how to differentiate between valid and invalid arguments through a solid understanding of logic.

Real-world applications:

As our world grows increasingly dependent on technology, our students are more susceptible than ever to fall prey to fake news, deep fakes, and whatever else the internet comes up with next. **Having the skills and abilities to tell the difference between fake news reports and real ones is crucial for our students now more than ever.**

Peer pressure and the need to fit in is pressing on our students through the increased use of technology. Not only are their friends putting pressure on our students, but so do the phones in their pockets. Thinking logically often helps students to see past the liminal experiences of the present and think critically about the consequences of their actions. Exposed to literally thousands of advertisements, videos, and unrealistic expectations, students must be trained to think logically in order to stay true to themselves and stay away from unhealthy habits promoted through social media.

What are the implications for academic success?

As educators, we all hope for our students to reach their utmost academic potential. One way that we, as educators, can boost our students likelihood of achieving academic success is teaching them

how to think logically. **Simply igniting that critical thinking aspect allows our students to overcome the adversaries that stand in the way of academic progress.**

Reading is more than just words

A student's probability of completing high school can be determined based on their grade three reading abilities. When measuring students' reading sufficiency there is more taken into account than just being able to sound out the words on a page. Comprehension as well as response are equally as important when it comes to reading. If you cannot understand the sentence you have read and respond accordingly there is not much point in having the ability to read the words. What do reading abilities have to do with logic you might ask? **Comprehension and response are learned through logical thinking.** Being able to think deeply about the text at hand will drastically increase a students reading sufficiency.

Library Resources

What is in the RPL "Let's Go Code" class-set box?

Activity 2: Coding Maze Mat-ness

Coding Maze Mat-ness

Project Description

Before we dive right into coding on the computer, let's introduce coding kinesthetically! In groups, "code" your friend's gross motor skills so they can save the robot through these coding mats. Jump, turn, and step until each mission is completed!

Big Ideas

Help students understand the basics of coding before they even open a computer. Coding is a language used to communicate with computers, and in this activity students will be using coding language to communicate with each other.

Students will become accustomed to logical thinking patterns as they work towards a bigger picture. This bigger picture emphasises order and as well as the planning of routes and directions.

Materials

no consumable materials needed for this lesson!

Equipment

Let's Go Code!™ Activity Set

Procedure

Introduction to Coding and Class Demo

- Discuss what coding is and how we are going to code without using a computer
    - "Coding is like a language that tells the computer or machine what to do. Once we know how to use coding languages, we can create video games, programs, apps, music, art, etc. to satisfy our creative desires. Instead of throwing you kids right into coding on the computers, we're gonna move around to teach you about the building blocks of coding. This is important for critical thinking, sorting information, and mapping routes b/w endpoints."
- Introduce the mats, coding cards and activity
    - Begin by creating this maze in the front of the classroom (make sure the kids can see):

    - The goal is to start at the arrow (stand on the arrow and face the direction the arrow is pointing at) and to get to the robot using the coding cards
    - Introduce the coding cards (forward and turn cards) and as a class, determine the order of coding cards needed to get to the robot (for each attempt, have an instructor or JC move through the maze while the class calls out the commands. If it is incorrect, troubleshoot and try again!):
- Split the campers into groups of 4 (or 2) and hand each group a "Let's Go Code Activity Set"

- 1st camper creates this maze: 
- 2nd camper creates the code for the maze: Forward, turn, forward, forward, turn, forward
- 3rd camper and 4th camper tests out the code and the maze

- Once the first maze is completed, have campers include the X card and a gear card onto the maze. Introduce the jump coding card and the pick-up coding card:  
  -  the X card: Blocked! You can't step on this mat. You need the jump (jet-pack) coding card to get over this.
  -  (gear card) or  (spring card): Pick up and bring to the robot for repairs with the "pick-up" (claw) coding card
  - Have campers switch roles and create the coding card sequence to get to the robot while overcoming the X card and gear card the placed.
- Once a group completes step 4, hand out the mission sheet with the different mazes and have each team complete each one.
  - Each team member will be switching roles for every new maze they do!
  - A team has completed a maze when the group performs the code (campers calling out the coding card commands and other campers performing it on the maze) **in front of the instructor**
  - Once the instructor has 'passes' the group, they can continue to the next maze
- [EXTENSION] Have campers create their own mazes and coding card sequences and have other groups try each other's creation
- CLEAN UP: Make sure you collect each piece from the Let's Go Code! activity set. Please notify your public library branch if the pieces begin to show signs of wear-and-tear.

MAZE MISSIONS

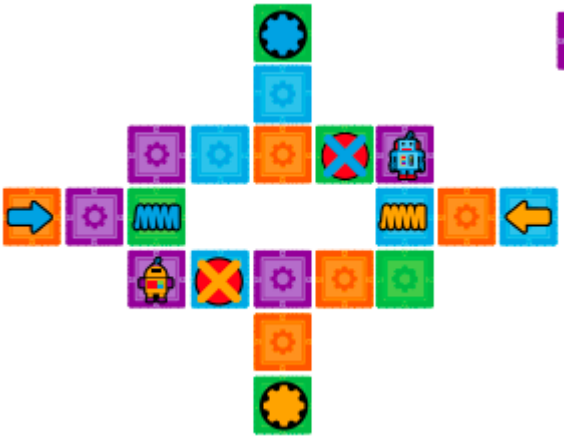
DEMO:

Mission 1 	**Mission 2** 
Mission 3 	Mission 4 (Difficult) 

Mission 5 (Difficult)

Mission 6 (Requires group of 4)

Mission 7 (Difficult - Requires group of 4)



Follow up

Have your students reflect on how coding might be different with a computer instead of a friend. How might the task be easier? How might the task be more challenging?

Ask students if they could imagine using this protest to create a video game or a robot?

Resources

https://vidweb.aws.marketlive.com/learningresourcesuk_vid/images/set_a/en_us/global/globalgraphics/LER%202835%20-%20Coding%20Buddies%20(Guide).pdf