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 1: If Else Simon!

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:

1

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

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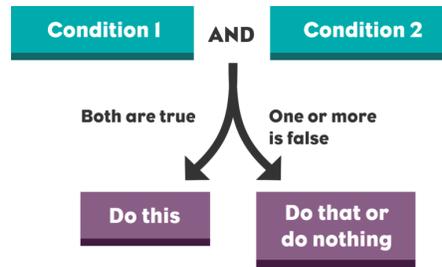
Conditional Statements
- **If/Else**- the "if" statement tells the computer what to do if the condition is true. The "else" statement tells the computer what to do if the condition is false
  - ex. *If* the number is divisible by 2 it is even, *else* it is odd.
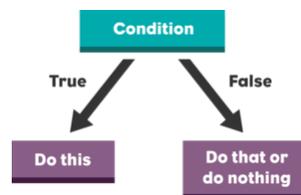

  - Multiple conditions can be added by inputting an additional else if statement between the initial conditions
  - ex. *If* you are free tonight we can meet then, *else if* we can meet tomorrow *else* we can meet on Saturday.
  - conditions are tested from the top to bottom
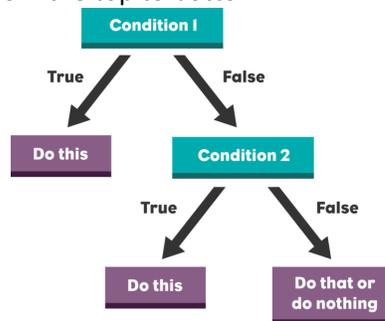

- **Logical Operators:** combine many boolean statements together (Checkpoint: **Boolean statements** are statements that are either true or false)
  - three main operators: and, or, & not.
  - **And:** for an "and" statement to be true, all of its' conditions must be true
    - True and True = True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

2

- **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
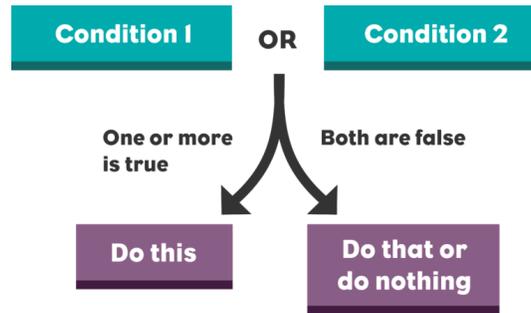
- **Not:** switches the value of the boolean to be the opposite of what it is
  - Not true = false
  - Not false = true
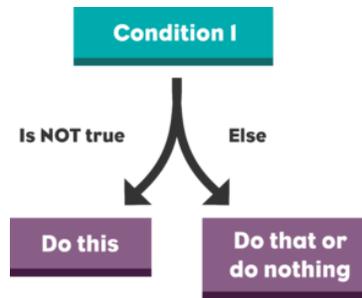
- **For Loops-**repeat a block of code a specified number of times. You determine how long your loops will go *for*
  - often use a variable, defined as counter, inside your loop to count how many times your loop has been run

3

```
Start Loop               Start Loop ➡
number = 1               Number = 1
                         Alert: "hello1"
    ↓                    Number = 2
                         Alert: "hello2"
  Alert                  Number = 3
"Hello" + "number"       Alert: "hello3"
                         Number = 4
                         Alert: "hello4"
  Add 1 to               Number = 5
  number                 Alert: "hello5"
                         Number = 6
    ↓
                         ➡ End loop
if number > 5
Exit Loop
```
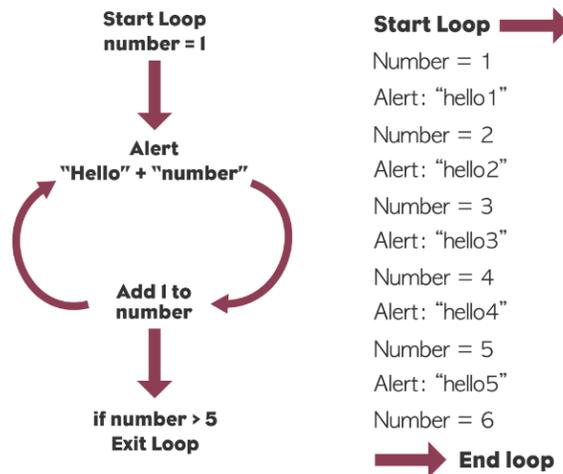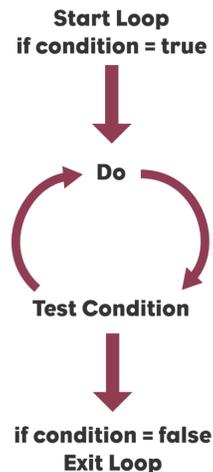
- **While Loops:** loops that will continue to go until the condition(s) are no longer true
  - ex. **while** (amount of people at a party) > 0, **do** play music.
  - As demonstrated above there are two parts to a while statement: while and do

```
Start Loop
if condition = true

    ↓

   Do

Test Condition

    ↓

if condition = false
Exit Loop
```

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

4

- 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

5

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 1:

If Else Simon!

Project Description

If you play this game, *then* you'll learn about computer coding logic and laugh a lot, *else* you'll miss out on a lot of fun!... In computer coding, instructions are written in the format of "If… Then… Else…". This is known as 'Programming logic'. Even basic, drag-and-drop block coding (such as coding Microbits and Ozobots) requires an understanding of these logic principles. In this unplugged Simon-Says style game, students can practice and get familiar with programming logic!

Big Ideas

Conditionals in coding:

Computer programs are made up of instructions that tell the computer how to process input and deliver output. The instructions you write are triggered/activated by events or commands. An important part of programming is telling the computer WHEN to perform a certain task. For this, we use something called 'conditionals', because a certain Condition or Rule has to be met before an action is performed.

6

Students are already familiar with the concept of conditionals in their daily lives. Have they ever had an adult say any of the following to them?

"If you clean your room, you can go out with your friends."

"If your homework is done, you can play video games."

"If you do your chores all week, you get your allowance, or else you are grounded."

These are all conditionals!

Conditionals follow the format of IF this, THEN that.

IF (condition is met), **THEN** (action performed)

Moreover, conditionals follow the format of IF this, THEN that, or ELSE that.

IF (condition is met), **THEN** (action performed), **ELSE** (different action performed)

For example, for the "IF it is snowing, THEN wear boots, ELSE wear shoes" conditional, notice that without the ELSE action, your students might end up not wearing any footwear!

Safety Considerations

Give students plenty of space since they may be moving their limbs vigorously in random directions. Let's avoid black eyes if we can!

Materials

No materials needed

Equipment

1 designated person as "The Simon" to give the commands (preferably the teacher/instructor).

Procedure

1. Encourage the entire class to stand up and follow the instructions that you give them as The Simon.
2. Begin with only "If…" statements. For example, "If you went swimming this weekend, do the chicken dance".
3. After the class is comfortable, move onto "If… Then… Else…" statements.
4. After the class is comfortable, move onto "If… Then…Else If… Else"

a. For example, "**If** you like pepperoni, then spin in a circle, **else if** you like ham sit on the floor, **else** do star jumps".
   b. This condition gives students two "**if"** options. If they don't agree on either option, they follow the "**else"** command.
5. After the class is comfortable, move onto "While… If… Then… Else…".
   a. The while statement gives a condition that must be met before the students can follow through with the "If... then... else…" command.
   b. For example, "**While** my hand is on my hip (this is the Simon saying this), **if** you can skateboard, **then** shake your friends' hands, **else** cross your arms".
   c. The Simon might say this condition but pause for 10 seconds, and then put their hand on their hip. At this time, the students are then able to do the command.
6. Now mix 1. to 5. together!

Examples of questions to ask:
- "IF you have brown hair, THEN raise your right hand, ELSE sit down."
- IF your name starts with a 'J', THEN give me a thumbs up
- IF the month of your birthday ends with a 'Y' or an 'R', THEN raise both your arms
- IF you play soccer, THEN kick your feet (be careful not to kick anyone)
- IF your favorite ice cream flavor is chocolate, THEN stick out your tongue, ELSE make rabbit ears behind your head
- IF you play a musical instrument, THEN snap your fingers, ELSE whistle
- IF your favorite monster in Minecraft is a zombie, THEN make zombie noises and movements, ELSE jump up and down
- Other example conditions
- Your hair is brown / blond / red
- Your eyes are brown / blue / green
- You are wearing a sweater / shorts / T-shirt / pants / skirt / sneakers / sandals
- It is raining / sunny / cloudy today
- You've played soccer / volleyball / football / frisbee in the past month
- You've eaten spaghetti / cereal / ham / toast in the past week
- You slept well last night / had a hard time waking up this morning
- Your favorite ice cream flavor is chocolate / vanilla / strawberry / mint / cookie dough
- You have a pet dog / cat / bird
- You have a sister / brother / no siblings

Get creative, and be silly!

Follow up

Since most of the command responses require physical movement it is easy to judge if the students are understanding the principle of conditions.

8

Extensions and Accommodations

- Lead the activity to practice conditionals with simple instructions, yet visually different physical poses.
- Verify that students have correctly executed the instructions based on the criteria.
- Read the conditional statements **slowly** and **clearly**.
- Make sure students understand and are following along with each set of statements.
- Reset the students to normal standing positions between each conditional statement.
- Ask the class for a few volunteers to create their own conditional statements for the rest of the class to follow along.

Future...

Use Micro:Bit MakeCode to put your logic into action!

Resources

This activity has been adopted from
https://minecraft.makecode.com/courses/csintro/conditionals/unplugged