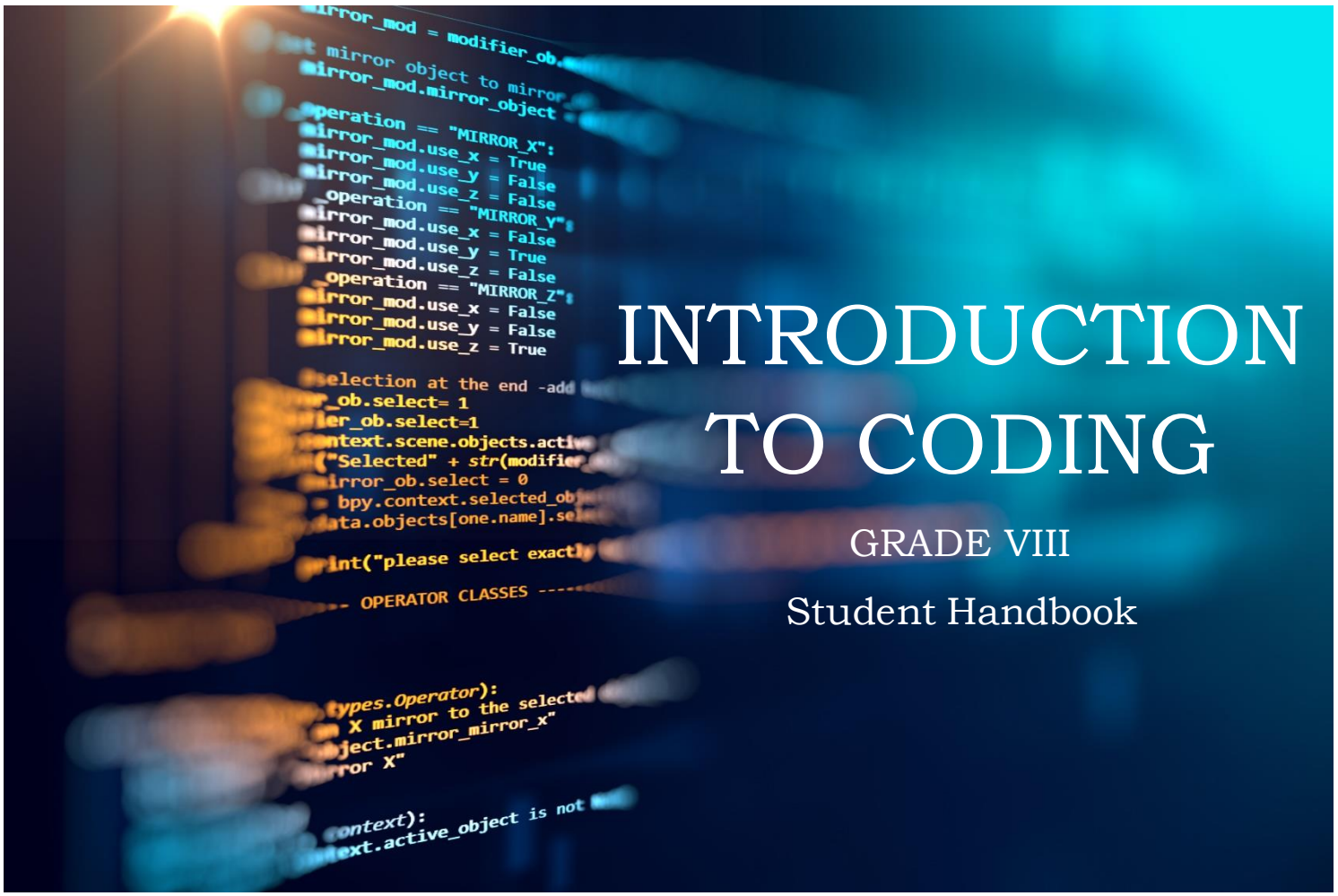


INTRODUCTION TO CODING

GRADE VIII

Student Handbook

Version 1.0

Microsoft

INTRODUCTION TO CODING

GRADE VIII

Student Handbook

ACKNOWLEDGEMENT

Patrons

- Sh. Ramesh Pokhriyal 'Nishank', Minister of Human Resource Development, Government of India
- Sh. Dhotre Sanjay Shamrao, Minister of State for Human Resource Development, Government of India
- Ms. Anita Karwal, IAS, Secretary, Department of School Education and Literacy, Ministry Human Resource Development, Government of India Advisory

Editorial and Creative Inputs

- Mr. Manuj Ahuja, IAS, Chairperson, Central Board of Secondary Education

Guidance and Support

- Dr. Biswajit Saha, Director (Skill Education & Training), Central Board of Secondary Education
- Dr. Joseph Emmanuel, Director (Academics), Central Board of Secondary Education
- Sh. Navtez Bal, Executive Director, Public Sector, Microsoft Corporation India Pvt. Ltd.
- Sh. Omjiwan Gupta, Director Education, Microsoft Corporation India Pvt. Ltd.
- Dr. Vinnie Jauhari, Director Education Advocacy, Microsoft Corporation India Pvt. Ltd.
- Ms. Navdeep Kaur Kular, Education Program Manager, Allegis Services India

Value adder, Curator and Co-Ordinator

- Sh. Ravinder Pal Singh, Joint Secretary, Department of Skill Education, Central Board of Secondary Education

ABOUT THE HANDBOOK

Coding is a creative activity that students from any discipline can engage in. It helps to build computational thinking, develop problem solving skills, improve critical thinking and exposure to real life situations to solve problems in various realms.

Therefore, CBSE is introducing 'Coding' as a skill module of 12 hours duration in classes VI-VIII from the Session 2021-2022 onwards. The idea is also to simplify the coding learning experience by nurturing design thinking, logical flow of ideas and apply this across the disciplines. The foundations laid in the early years will help the students to build the competencies in the area of AI, data sciences and other disciplines.

CBSE acknowledges the initiative by Microsoft India in developing this coding handbook for class VIII students. This handbook uses block coding to explain advanced concepts of coding using python in MakeCode platform. It uses gamified learning approach to make learning experience more engaging. The book is intuitive with practical examples of theoretical concepts and applied exercises. There are mini projects that students can work on. Additionally, the handbook also focuses on creating exposure to ethics of coding and application of coding in other subjects like mathematics.

The purpose of the book is to enable the future workforce to acquire coding skills early in their educational phase and build a solid foundation to be industry ready.

RESOURCES FOR STUDENTS

Minecraft education edition

Minecraft education edition is a game-based learning platform that promotes creativity, collaboration, and problem-solving in an immersive digital environment. This platform provides a fun way of learning coding and design thinking concepts.
Visit https://education.minecraft.net/ for more details.

MakeCode

Microsoft MakeCode is a free, open source platform for creating engaging computer science learning experiences that support a progression path into real-world programming. It brings programming to life for all students with fun projects, immediate results, and includes both block and text editors for learners at different levels.
Visit https://www.microsoft.com/en-us/makecode for more details.

GitHub

GitHub is a storehouse where you can manage and collaborate on your code. It helps to maintain different versions of the code easily. GitHub Student Developer Pack gives students free access to the best developer, web development, gaming and many other tools at no cost enabling practical learning.
Sign up for the GitHub Student developer pack here
https://education.github.com/discount_requests/student_application?utm_source=2021-06-11-cbse

TABLE OF CONTENTS

ETHICAL PRACTICES IN CODING

As you build capabilities around coding, you will be equipped to build software on your own which will have an impact on society in general. So, it is very important to adhere to ethical practices while building your own code. Below are some practices you must remember as you keep learning to code.

Respect other's privacy

As you keep building software of your own you might need to access other's personal information. Make sure you do not misuse the data and you understand your rights and responsibilities associated with that data

Honor Confidentiality

You might be given access to confidential information like trade secrets, client data, financial information etc. Ensure you do not disclose this information without proper authorizations.

CONDITIONALS IN DETAILS

1.1 What will you learn in this chapter?

In this chapter you will understand conditionals in detail and how to use conditionals in your code.

Here you will learn:

- Understanding IF ELSE and ELSE-IF statements
- Using logical operators
- Using nested conditions

1.2 Types of Control Structures

In programming, control structure is a block of program that accepts and analyses different variables and chooses the direction in which to go based on the given parameters. To put simply, it is a basic decision-making process in computing. It is like a prediction that program makes by accessing the given parameters.

There are three basic types of control structures in programming:

1. Sequential
2. Selection / Conditional
3. Iteration

Sequential - In a sequential control structure, the statements in a program are executed sequentially, i.e. step-by-step in an order that they are written.

Selection / Conditional - A selection (conditional) control structure is used to test a condition in a program. This control structure takes a decision to execute one statement/operation over another statement/operation depending on the condition.

Iteration – This control structure executes a set of statements for a certain number of times till the mentioned condition is true. Loops are examples of iterative statements.

Using this control structures, programming has derived many types of control statements to suit different needs. We will study them in upcoming topics.

```
a = 34
if a % 2 == 0:
    print("The number is even")
else:
    print("The number is odd")
print("Here it is anyway")
```

1.3 Understanding IF-ELSE and ELSE-IF statements

Every day, we take many decisions depending on our situation. For example, we decide whether a number is even or odd by dividing the number with 2. If the remainder is 1, we can say the number is odd.

IF-ELSE Statement

If we want to implement the logic in our program, we need to use **if-else** statement.

The **if-else** statement requires three things.

1. Evaluation statement: condition or expression that is being checked
2. Execution statement: the operations that will be performed if the condition appears to be true
3. Else execution block: the operation that will be performed only if the evaluation statement is false

If we need to determine, a number is even or odd using a flowchart, it becomes like the following diagram.

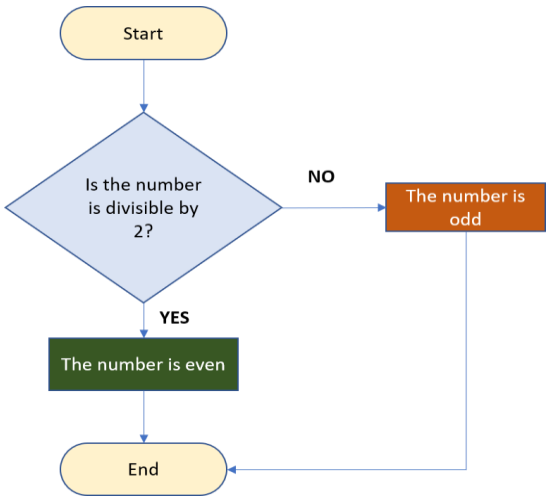


Fig 1.1 *Understanding if-else*

If we implement the flowchart using coding syntax, it will be like the following lines of code.

In above example ***a %2 == 0*** is the expression that we are using for evaluation, and in the next line we are using indentation to indicate the execution statement. In the next lines we are using **else** keyword to indicate what statements will be executed in case the expression used in false. The last line of code will be executed always because we are back using the origin indentation.

If we implement the scenario using proper syntax it will look like the following lines of code.

```
a = 34
b = 34
if a > b:
    print("a is greater than b")
elif a == b:
    print("a is equal to b")
else:
    print("a is less than b")
```

Point to note over here is that "=" is an assignment operator, i.e. it is used to assign a value to a variable. And "==" is a relational(comparison) operator, i.e. it is used to compare the value of two variables.

ELSE-IF Statement

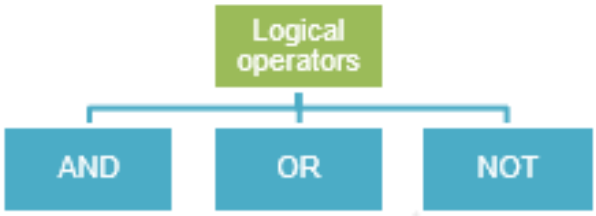
Let's consider a situation where we need to compare two numbers a, b and determine if a is greater than b or they are equal. To implement this one, we need to perform checks at two steps. The keyword we need to use is elif. The keywork elif is used for checking another statement in case previous statement is false.

1.4 Logical Operators

Logical operators are fundamental blocks that can be used to build a decision-making capability in your code. When we need to make our decision based on two or more checks, it is necessary to have a combination of logical operators.

We can do things conditionally in our programs using **if statements** or **if/else statements** combined with logical operators. Logical operators work like Boolean variables and return either TRUE or FALSE.

The three most important logical operators are AND, OR and NOT.



AND Operator

The AND operator is used to see if two or more conditions are true. If all the conditions are true, the AND operator returns TRUE. If any one of the conditions fail, the AND operator returns FALSE. In python AND operator is denoted by **and** keyword. Some other programming languages use "&&" as AND operator.

OR Operator

The OR operator is used to see if either one of two or more conditions is TRUE. If any of the condition is true, the OR operator returns TRUE. If all the conditions fail, the OR operator simply returns FALSE. In python syntax OR operator is denoted by **or** keyword. Some other programming languages use "||" as OR operator.

NOT Operator

We use the NOT operator to reverse or negate a condition. If the condition is true, NOT will return false and vice-versa.

In python programming, we use **not** keyword to denote not operator. Some other programming languages use "!" as NOT operator.

Relational operators

Let us now look at the full list of relational operators.

Operator	Symbol	Example	Meaning
Greater than	>	x > y	x greater than y
Equal to	==	x == y	x is equal to y
Less than	<	x < y	x is less than y
Greater than or equal to	>=	x >= y	x is either greater than or equal to y
Less than or equal to	<=	x <= y	x is either less than or equal to y
Not equal to	!=	x! = y	x not equal to y

Point to note over here is that "=" is an assignment operator, i.e. it is used to assign a value to a variable. And "==" is a relational(comparison) operator, i.e. it is used to compare the value of two variables.

Activity: Can you create a triangle?

Take this example of following figure of triangle with sides A, B and C. We know

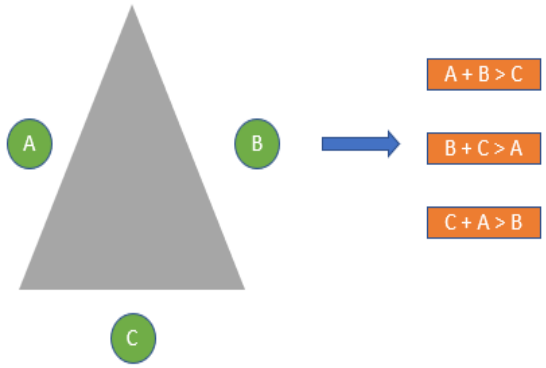


Fig 1.2 *Can you create a triangle.*

the sum of two sides must be greater than remaining side.

Unless all sides are compared, we cannot say values for A, B and C are valid or not. Therefore, we need to combine all three logical expressions to say whether they can truly make a triangle.

Now, if we want to solve the problem of triangle in python coding it becomes as following.

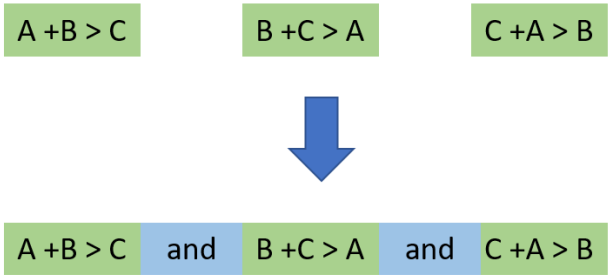
```
a = 10
b = 12
c = 16

if (a + b >c ) and (b+c>a) and (c+a>b):
    print("Yay! triangle is possible")
```

Fig 1.3 *Conditions to create a triangle.*

1.5 Precedence of Logical Operators

Just like arithmetic operators, logical operators also have precedence that determines how the operations are grouped in the absence of parentheses. Parentheses can be used to group the operands with their correct operator, just like how we do it in arithmetic.

In an expression, the operator with highest precedence is grouped with its operands first, then the next highest operator will be grouped with its operands, and so on. In a situation where there are many logical operators



of the same precedence, they will always be examined from left to the right.

Have a look at the below table to check the precedence of logical operators:

Operator	Precedence
NOT (!)	High
AND (&&)	Medium
OR (||)	Low

In the below example, A1, A2, A3 and A4 stand for the relational expressions. They have a mix of "&&", "||" and "!" operators. The expression in the first column can be interpreted by parentheses as shown in the third column.

It is a common practice to use parenthesis to group operands together rather than relying on logical operator precedence rules.

A1 && A2 && A3 || A4	can be interpreted as	((A1 && A2) && A3) || A4
A1 || A2 && A3	can be interpreted as	A1 || (A2 && A3)
! A1 && A2 || A3	can be interpreted as	((!A1) && A2) || A3
A1 && A2 || A3 && A4	can be interpreted as	(A1 && A2) || (A3 && A4)

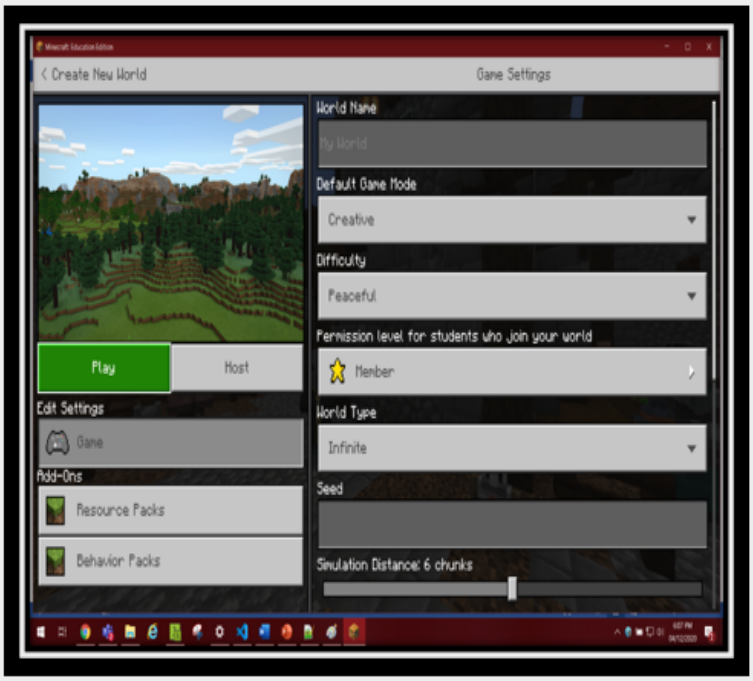
1.6 Activity: Display if a number is odd or even

Let us now run through a practice exercise to understand logical operations. You should try this exercise on the Minecraft Education Edition. You may set up the Minecraft education edition from https://education.minecraft.net/get-started

Note: Minecraft is just one of the platforms to achieve this output. You can use many similar platforms available online to achieve similar output like – Scratch (https://scratch.mit.edu/) and Code.Org (https://code.org/)

At the end of this activity, you can expect following output on screen:

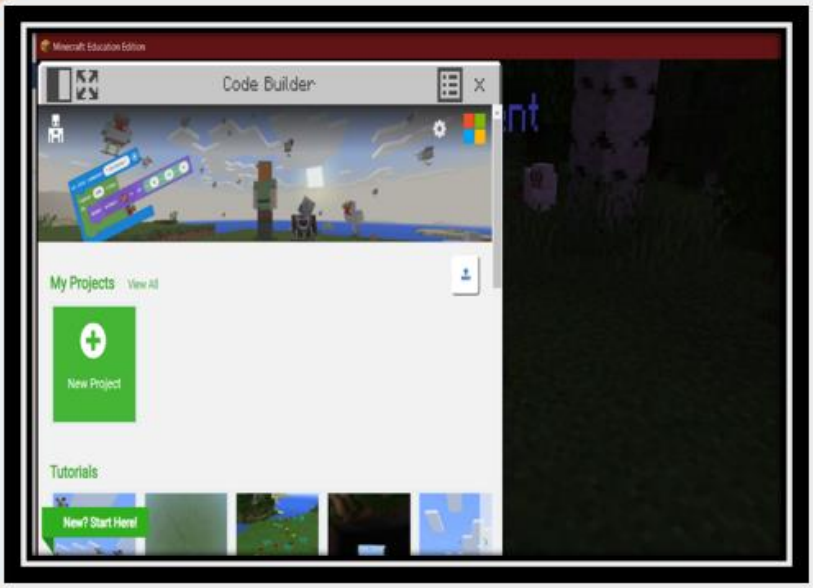
Let us now follow below steps to replicate this output on screen:



A new screen will appear, Click on **Create New** button as highlighted below, and from options make sure to click on **New** option. This will take to the **Create New World** screen. In this screen you will get to create new Minecraft world for yourself.

From the Create New World screen, make sure to choose Default Game Mode option as Creative. When you are done with that, click on the big Play button

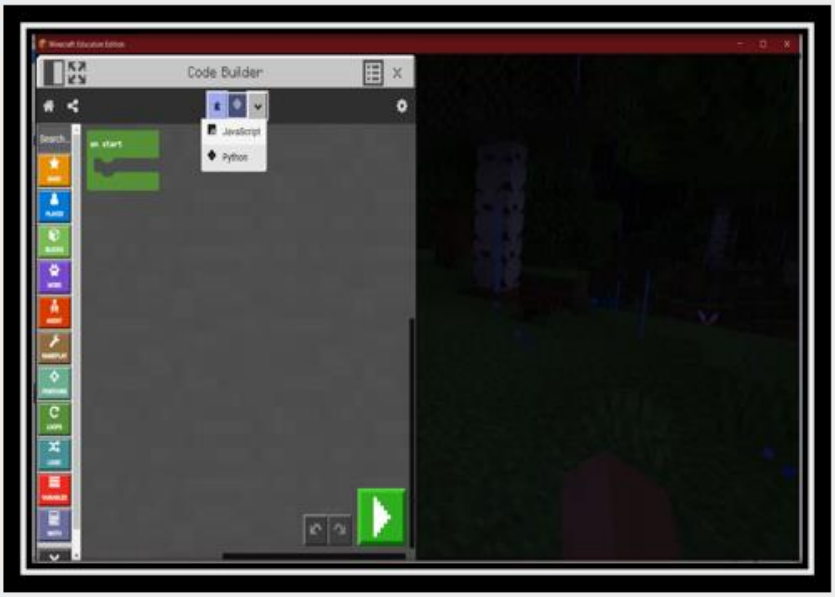
Minecraft Code Editor
Open Minecraft Education Edition application, login and click on **Play** button



Press Key **C** to bring up code editor. A screen will pop up asking you to create new project. Click on **New Project** button, give it a name.

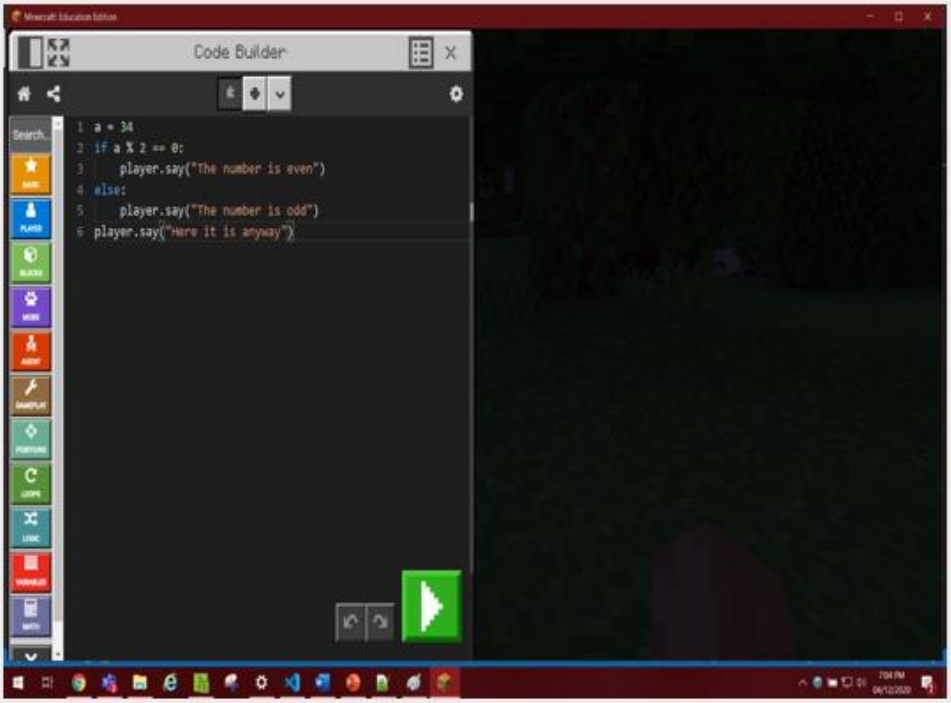
You will arrive to the Minecraft world that you have created

Press **Key C** to bring up the code editor



From code editor click on the highlighted dropdown button and select the option **Python**. You might see some code, make sure to clear all that

Now we will implement the example of odd-even to display some messages on the screen.



Now in the same screen you will be able to type lines of code. Now type the lines of code as given below. Make sure you do not make any spelling mistake and make the indentation as it is in the given picture.

The program that you have written assigns 34 to a variable named as a. Then, in the next lines of code, it checks whether variable a is now divisible by 2. If it is divisible, we are displaying a message on the screen. At line 3, 5 and 6 the statement **player.say** is used for displaying the message in the screen.

When you are done with typing the necessary lines of code, click on the green **Start** button to execute the program. You will see the below result displaying number is even.

Activity 1.1
Create a program in Minecraft which will check two numbers and display a message saying if one is greater than the other, or they are same.

1.7 Nested Conditional statement

In some cases, for checking complex scenarios we might find that using a single if-else loop is not enough. We can also use a combination of if, elseif statement inside another if block.

Suppose we want to check if a number is divisible by 4 or 5 or both 4 and 5. We need to check in multiple steps. First, we need to check if its divisible by 4, if this condition is true, then we further need to check if it is divisible by 5. In another IF block we again need to check divisibility by 5, and check if it is divisible by 4.

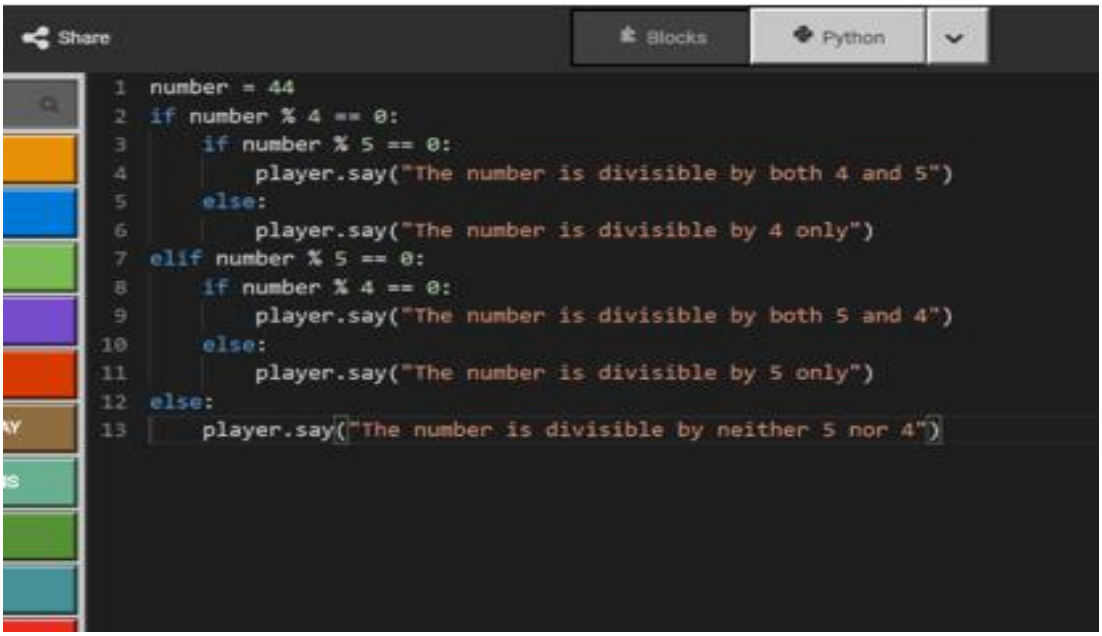
Syntax of nested conditional statements is like below

```
if condition 1:
   if condition 2:
      statement 1
   else
      statement 2
elif condition 3:
   if condition 4:
      statement 3
   else
      statement 4
else:
    statement 3
```

1.8 Activity: The nested division problem

We will now create the program to check if a number is divisible by 4 or 5 or both 4 and 5. To do that, open Minecraft Code Editor again. You can get the instructions from previous sections for doing that.

Now that you have Minecraft Code Editor opened, type down the program as given below. Make sure the indentations are correct.



After you are done with typing the code, click on the green **Start** button to execute the program.

Result:

You will now see that message is displayed according to the logic we have written in our program. It assigns 44 in a variable called **number**. Then it goes through our nested if-elseif blocks to

display the result. If you have typed the lines of code properly, clicking on **Start** button will display a message like below.

Now press C key to go back in the code editor. You can change the value of the variable and run it again. By doing that, you can verify if the program is behaving as you have expected.

Activity 1.2
Create another program which assigns total marks of a student to variable, and display grade using the logic below.

- Total marks > 70, then grade **A**
- Total marks >49 and Total marks <=70, then grade **B**
- Total marks <= 49, then grade **C**

1.9 Quiz time

Objective Type Questions

Question 1	Which one of the following options is not a logical operator?
Option 1	OR
Option 2	WHAT
Option 3	AND
Option 4	NOT

Question 2	Which of the following can create conditional statements?
Option 1	IF
Option 2	IF-ELSE
Option 3	Nested IF-ELSE
Option 4	All the above

Question 3	Which operator negates a condition?
Option 1	OR
Option 2	NOT
Option 3	AND
Option 4	None of the above

Standard Questions

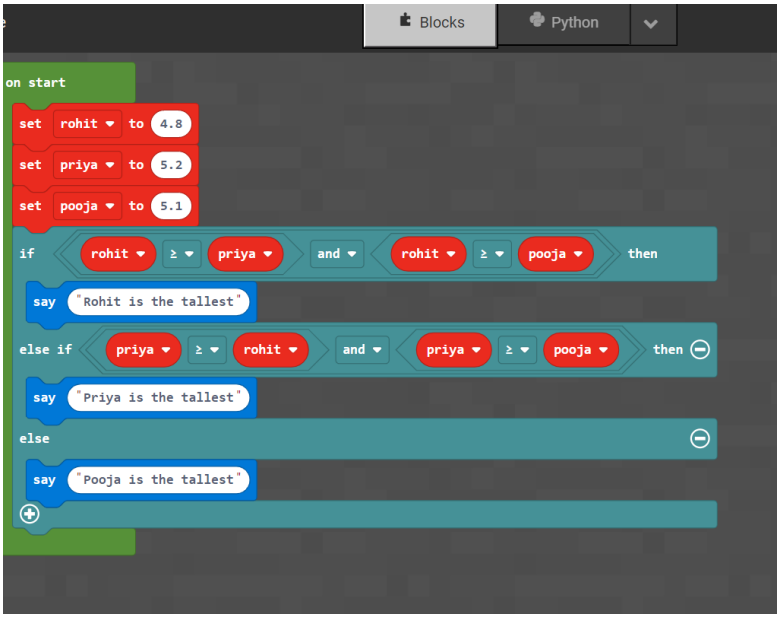
1. What are different types of control structures? Define each type.
2. What are the different types of relational operators?
3. What are logical operators? Name different types of logical operators.
4. What are the differences between "AND" and "OR" operators? Give examples.
5. What is a nested conditional statement and where is it used?

Higher Order Thinking Skills(HOTS)

1. Create a program in Minecraft which assigns a number to a variable. Then checks if the number is divisible by 7 or 9 or both 7 and 9.
2. Create a program in Minecraft which assigns a number to a variable. Then display if the number is even or odd.
3. Create a program in Minecraft which assigns a number to a variable. Find out if the number is divisible by 3. If divisible by 3, multiply it by 10 and display the output.

Applied Project

Problem Statement: There are three children named Rohit, Priya and Pooja. Height of Rohit, Priya and Pooja are 4.8 ft 5.2 ft and 5.1 ft respectively. Create a program in Minecraft using block coding to find who is the tallest of three.
Sample solution:



1.10 What you have learned in this chapter

- AND, OR NOT are the different logical operators
- We can combine multiple operators in a single expression
- IF-ELIF is used to write logical operators in python
- When we have an IF within another IF, we call it a nested conditional statement
- We learnt how to use operators in Minecraft code editor
- How to apply logical operators in python coding?

GET CREATIVE WITH LOOPS

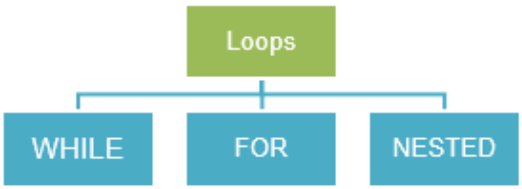
2.1 What will you learn in this chapter?

Now that we already understand logical operators in coding, let us dive deeper into the details of how to solve a coding problem. By the end of this chapter, you will learn:

- What are the different types of loops?
- Definition of different types of loops
- Break and Continue loops
- Get introduced to pseudocode

2.2 What are Loops?

Everyday there are some tasks which need to be done repeatedly. Loops are programming elements which repeat a portion of the code for a set number of times till the desired process is complete. Repetitive tasks are common in programming and loops are important to save time and minimize errors.



Loops make our code more manageable and organized. Let us now see what the different types of loops are:

1. While Loop
2. For Loop
3. Nested Loop

The While Loop

The While loop can execute a set of commands till the condition is true While Loops are also called conditional loops.
Once the condition is met then the loop is finished

For example - Print from 1 to 10

Here, if we want to derive the loop from this scenario, we have the following conditions:

Condition: Write from 1 to 10

And the decision we are deriving is:

Decision: Have we reached 10

Based on this we can write the below pseudocode:

```
x = 0
while x is not 10
  x = x + 1
  print(x)
```

The For Loop

For loop is needed for iterating over a sequence.
For example – We need to calculate the square of numbers present in a list.

Numbers= [1, 3, 5, 7, 13]

The pseudo code for this will look like below:

```
numbers = [1, 3, 5, 7, 13]
sq = 0
for x in numbers
   sq = x * x
   print(sq)
```

This is the result for the code

```
1
9
25
49
169
```

The Nested Loop

Loop can be nested in Python. A loop that occurs within another loop is called nested loop. Consider the below program.

```
numbers = [1, 2, 3]
alphabets = [a, b, c]
for num in numbers
    for alphabet in alphabets
        print(alphabet)
```

We have used two loops here. The outer loop, iterates over the numbers array and the inner loop iterates over the alphabet array. So each of the alphabets get printed 3 times.Finally, we have 9 items that are printed.

```
a
b
c
a
b
c
a
b
c
```

2.3 Exit Criteria

Now that we have understood about loops and its iterations, it is also important to understand when and where should one stop iterating through these loops. As mentioned in the previous topic, it is very important to keep in mind while programming that the looping condition should result false at a certain point in time. Otherwise, the block of code will enter an infinite loop. To make sure that the loop does not enter an infinite loop, it important to define an exit criterion for the loop.

Exit criteria is defined as a condition that must be met before completing a specific task. It is a set of conditions that must exist before you can declare a program to be complete. Exit criteria is one of the most important components while defining a loop. As without an exit criterion, the program tends to enter in an infinite loop. These criteria differ from program to program as per the requirement.

For example, while creating a loop to print numbers from 1 to 1000, exit criteria is that loop should exit the block of code when the 1000th number is

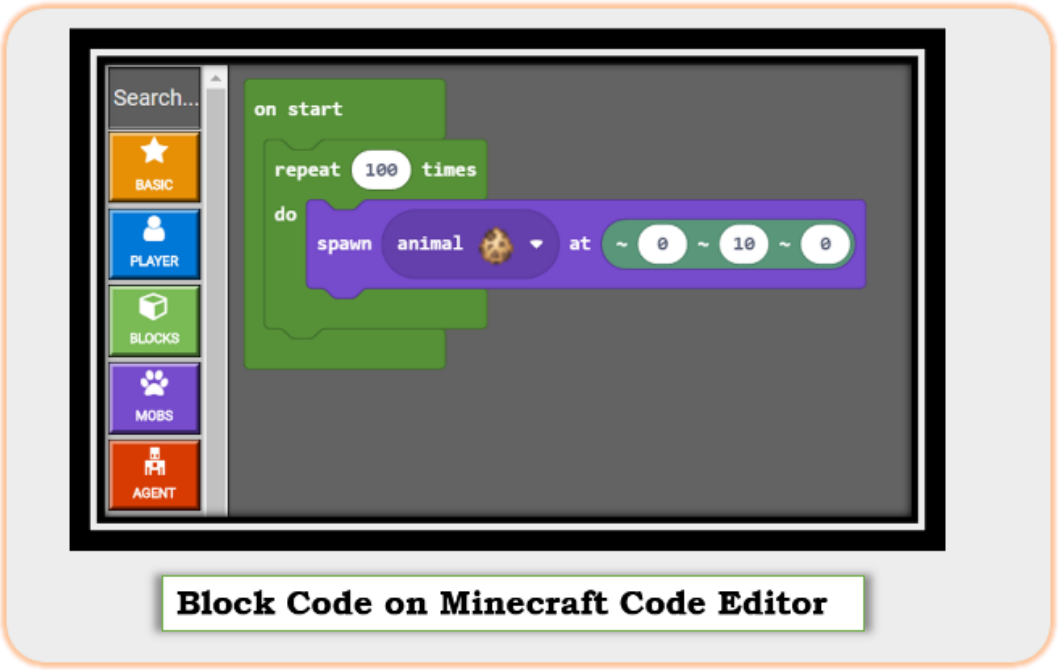
printed, else the program will enter an infinite loop.

Activity 2.1
Write a pseudocode to print prime numbers from 1 to 100

2.4 Activity: Cat Rain

You might have heard the phrase "raining cats and dogs". In this exercise we will see how you can rain cats from the sky. You should try this exercise on the Minecraft education edition.

By now you should already have an idea of block coding. Below is how the exercise will look when implemented with block coding with Minecraft

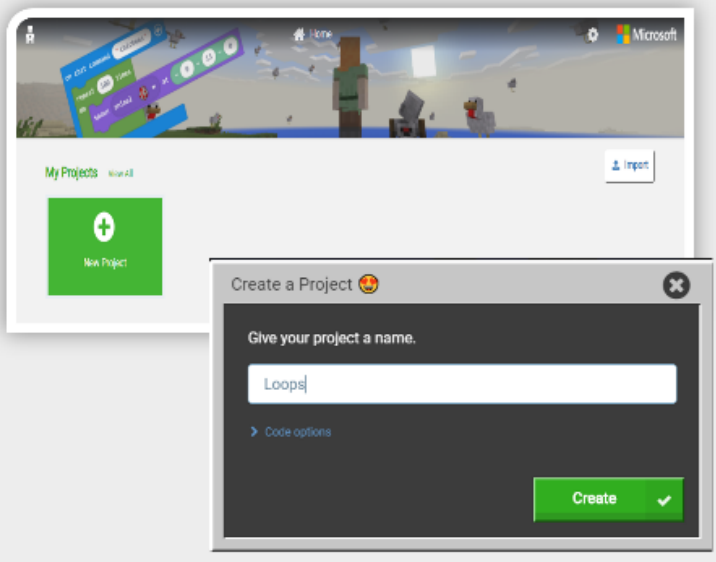


Block Code on Minecraft Code Editor

The final output of this exercise should look like as shown in the image below.

Final Output – Cat Rain Activity

However, in this exercise, we will try to implement the same code using python.



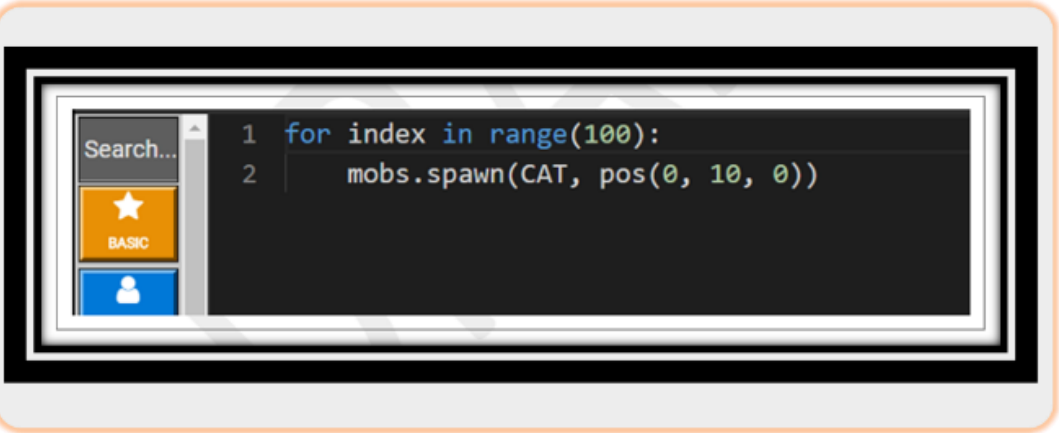
Creating New Project
You can create a new project by clicking on green box labeled as 'New Project'. A dialog box will appear prompting you to give a project name.

Giving Your Project A Name
You need to type down a name in the text and click on 'Create' button

Choose python as the editor

Follow the below code block. We have used a for loop with range 100 – This means that the loop will run 100 times.

So, the line **mobs.spawn(CAT, pos(0, 10, 0))** will be executed 100 times.



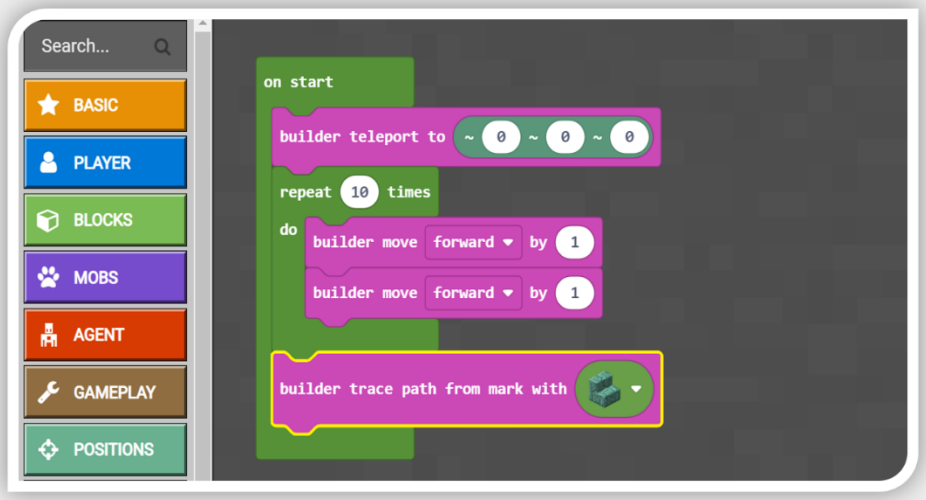
After you click on play 100 cats will rain from the sky.

Note: Minecraft is just one of the platforms to achieve this output. You can use many similar platforms available online to achieve similar output like – Scratch (https://scratch.mit.edu/) and Code.Org (https://code.org/)

2.5 Activity: Create a staircase using for loop

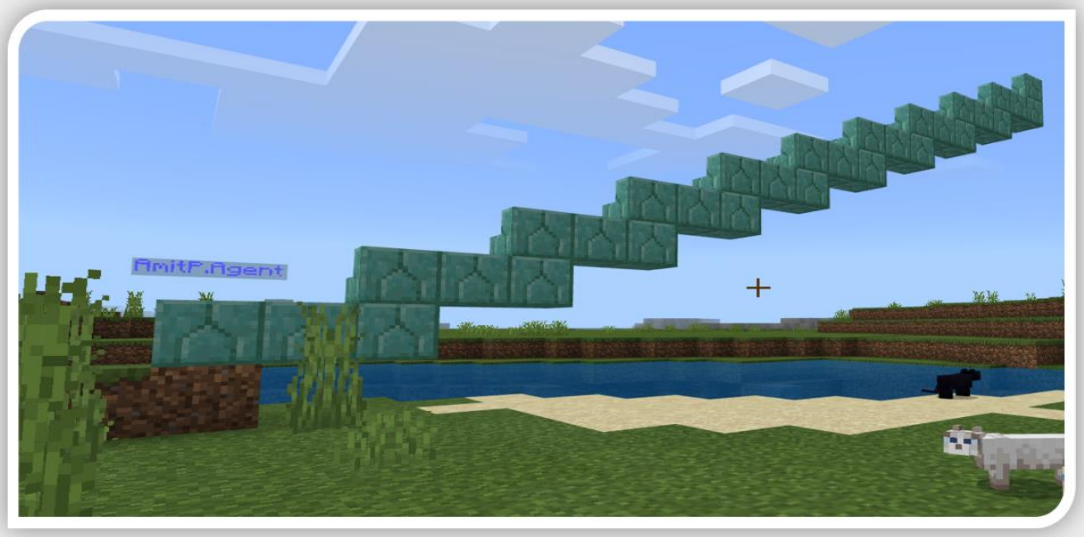
In this example we will create a block of stairs using tube coral black material where the for loop will run 10 times so the height of the stairs will be 10 blocks and the width will be three blocks.

Below is how the code will look when implemented with block code using Minecraft education edition.



Using block coding

At the end of this exercise, the final output should look like shown in the below image.



Final Output – Create staircase using for loop activity

```
builder.teleport_to(pos(0, 0, 0))
for index in range(10):
    builder.move(FORWARD, 2)
    builder.move(UP, 1)
builder.trace_path(TUBE_CORAL_BLOCK)
```

Using python

So, we see how the same action is performed using a for loop. The detailed explanation of the previous code is given below:

Step 1: Teleporting the builder to the specified position

Step 2: Creating a for loop which will run 10 times

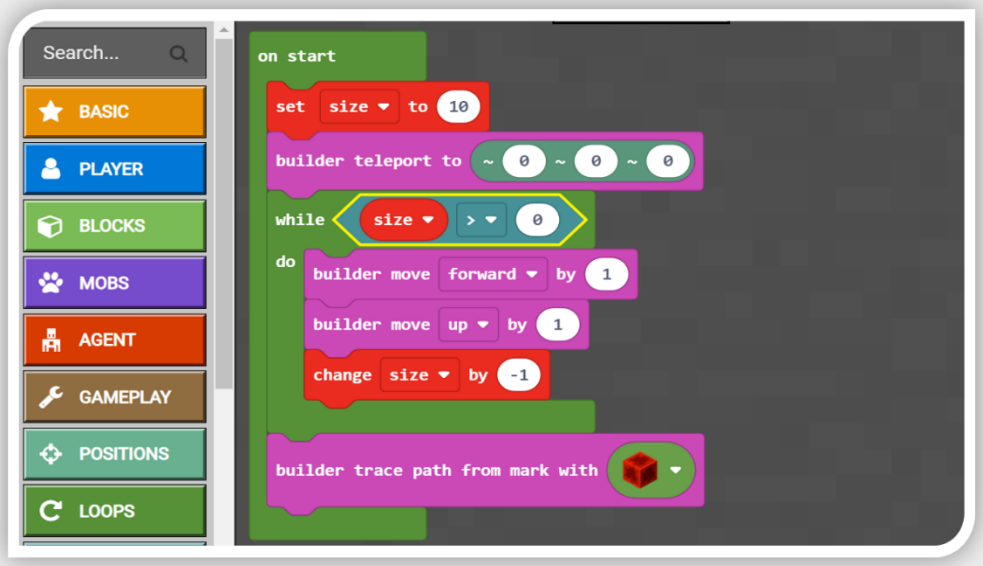
Step 3: Inside the for loop we will move the builder forward by 2 steps
Step 4: Move the builder up by 1
Step 5: Finally the builder traces the path travelled and puts a block on the position
Step 6: Click on play. You will see a flight of stairs of height 10 and having width of 3 blocks.

Note: Minecraft is just one of the platforms to achieve this output. You can use many similar platforms available online to achieve similar output like – Scratch (https://scratch.mit.edu/) and Code.Org (https://code.org/)

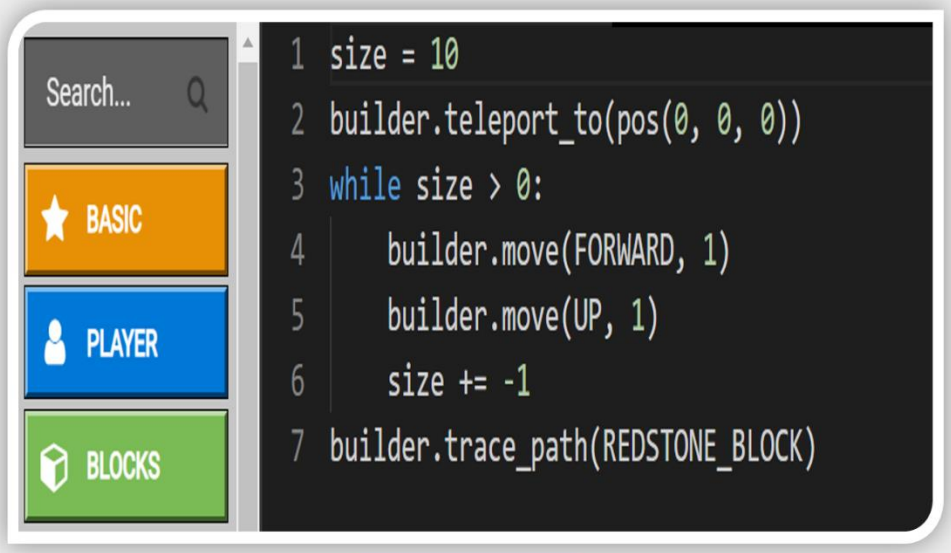
2.6 Activity: Create a staircase using while loop

In one of the previous activities we used for loop to create a staircase. Now let us create a staircase with while loop.

If we implement this with block coding, below is how the code will look

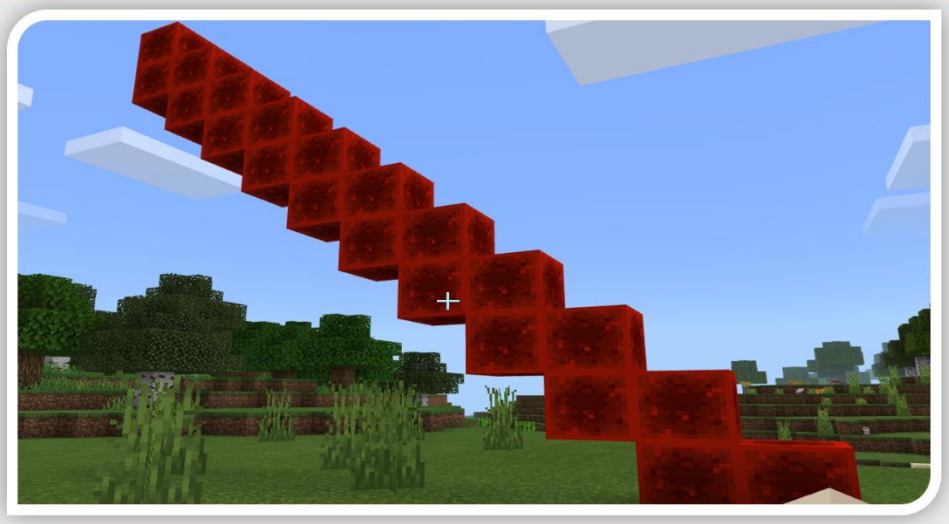


Using block coding



Using Python

At the end of this activity, the final output should look like shown in the image below.



Final Output – Create staircase using while loop activity

Below are the detailed steps:

Step 1: Set the value of size variable to 10

Step 2: Teleport the builder to a specified position

Step 3: Create a while loop of size greater than zero

Step 4: Inside while loop move the builder forward by 1 step

Step 5: Next, move the builder up by 1 step

Step 6: Update the size variable by size = size - 1

Step 7: Finally, the builder traces the path travelled and puts a block on the position

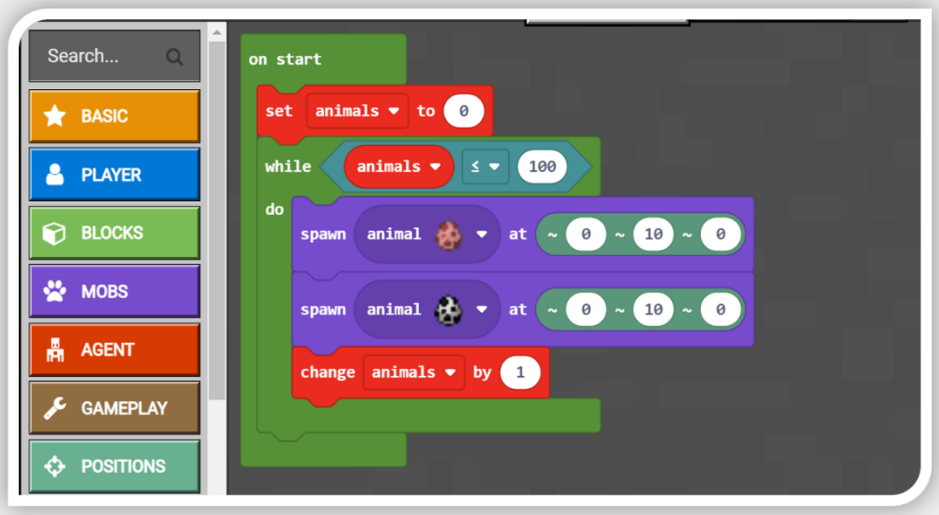
Step 8: Finally run the code by clicking on play then on the chat window type step 10

Results: You will see a stair of height 10 formed in your Minecraft world

2.7 Activity: Raining pigs and pandas

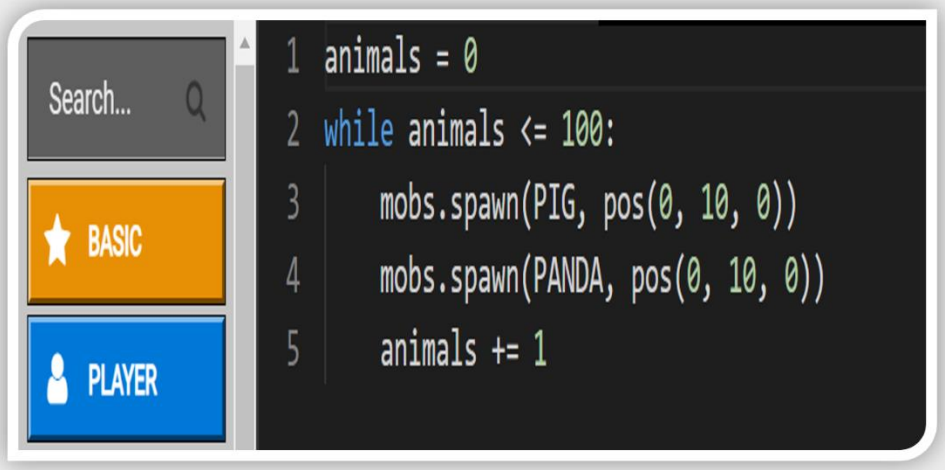
We have already gone through an exercise to rain cats using for loop. We will now try to take it further by raining Pigs and Pandas using while loop.

Below is how the code looks like when done using block code in Minecraft:



Using Block Code

And when done with Python, the code will look like below.



Using Python

At the end of this activity, the final output on screen should look like shown in the image below.

Final Output – Raining pigs and pandas activity

Below are the detailed steps:

Step 1: In the on start command, first set variable "animals" to 0.

Step 2: Now start a while loop which will run till the value of "animals" variable is less than or equal to 100.

Step 3: Inside while loop, set spawn of pigs to specified position.

Step 4: Inside while loop, set spawn of pandas to specified position.

Step 5: Lastly, update the value of variable "animals" to "animals = animals + 1".

Step 6: Finally run the code by clicking on play then on the chat window type rain 100

Result: After you run the code you will see 100 pigs and 100 pandas dropping from the sky in your Minecraft Environment

Note: Minecraft is just one of the platforms to achieve this output. You can use many similar platforms available online to achieve similar output like – Scratch (https://scratch.mit.edu/) and Code.Org (https://code.org/)

2.8 Quiz time

Objective Type Questions

Question 1	Which type of loop can be used till a condition is met?
Option 1	While loop
Option 2	For loop

Question 2	Which type of loop can be used to iterate over a sequence?
Option 1	While loop
Option 2	For loop

Question 3	Which statement is used to break the loop?
Option 1	Stop
Option 2	Break
Option 3	Exit

Standard Questions

1. What are the different types of loops?
2. What is exit criteria?
3. What is exit criteria for **While** loop?
4. What is exit criteria for **For** loop?
5. What are nested loops?

Higher Order Thinking Skills(HOTS)

1. Make 50 horses and 50 cats rain from the sky using for loop
2. Make stone stairs of height 15 blocks and width of 1 block using while loop

Applied Project

Problem Statement: Write an algorithm to find the HCF of two numbers using prime factorization method.

2.9 What have you learnt in this chapter?

- By now you should have understanding about loops. Loops are used to repeat a block of code
- Loops are also used to iterate through a sequence
- A loop within another loop is called nested loop
- Exit criteria is defined as condition which must be met before completing a specific task.
- Break statement is used to exit from a loop
- Continue statement is used to skip some lines of code within a loop

FUNCTIONS IN DEPTH

3.1 What will you learn in this chapter?

- How using function helps us?
- How to create and use functions in block coding
- Returning value from functions

3.2 Recap of functions

A function is a block of code made up of a set of steps that results in a single specific action. The programmer will give this action a simple name. Giving a simple name to a function, increases the chances that the set of steps can easily talked about and reused again and again in the program.

3.3 Function Parameters

By now, you already had a brief idea of what functions are and how it works. **Function parameters** are variables which are taken as inputs to the function to do a task when the function is called. An **argument** is the value passed to a function which is received by the function as parameter.

For example, if we need to calculate the volume of a cube, we need three variables length, breadth and height. The formula to calculate the volume of a cube is

where **V** is the volume of the cube, **L** is

V = L * B * H

the length, **B** is the breadth and H is the height of the cube. We can make a function named **VolumeOfCuboid** which takes parameters **L**, **B** and **H**. Now the function would look like

```
VolumeOfCuboid (L, B, H)
{
    return L * B * H
}
```

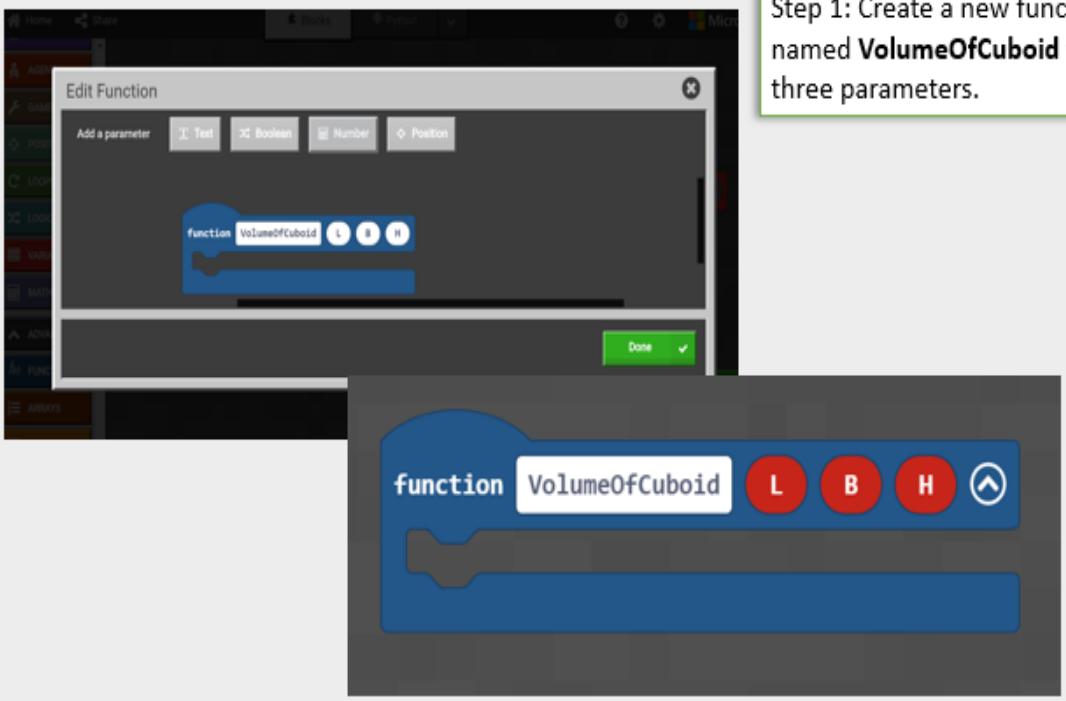
3.4 Activity: Calculating Volume of Cuboid

Let us now see how we can calculate the volume of a cuboid in Minecraft using the above steps. You should try this exercise on Minecraft using the Make Code editor for Minecraft which can be found here https://minecraft.makecode.com/

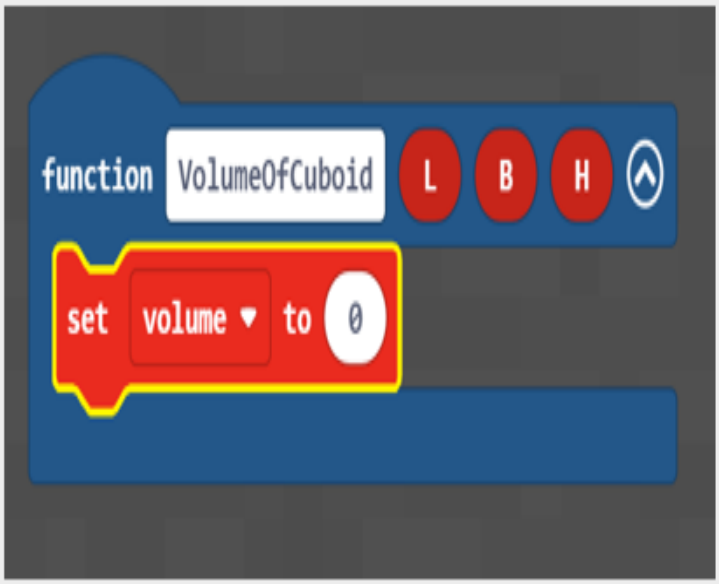
At the end of this exercise, the final output should look like shown in the screen below:

Final Output – Finding Volume of a Cuboid Activity

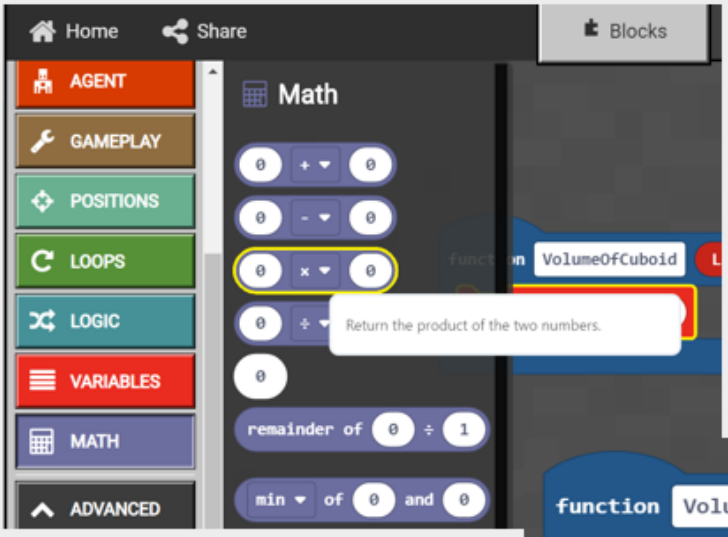
Let us now try to replicate this output on our screen by following the steps below:



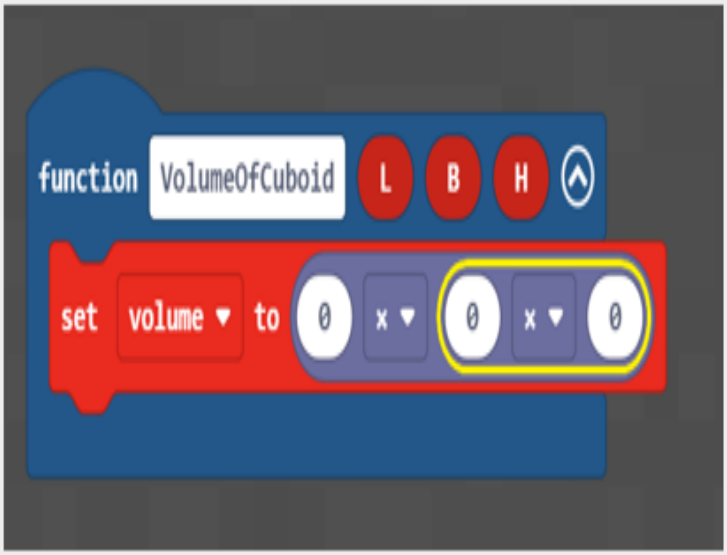
Step 1: Create a new function named **VolumeOfCuboid** with three parameters.



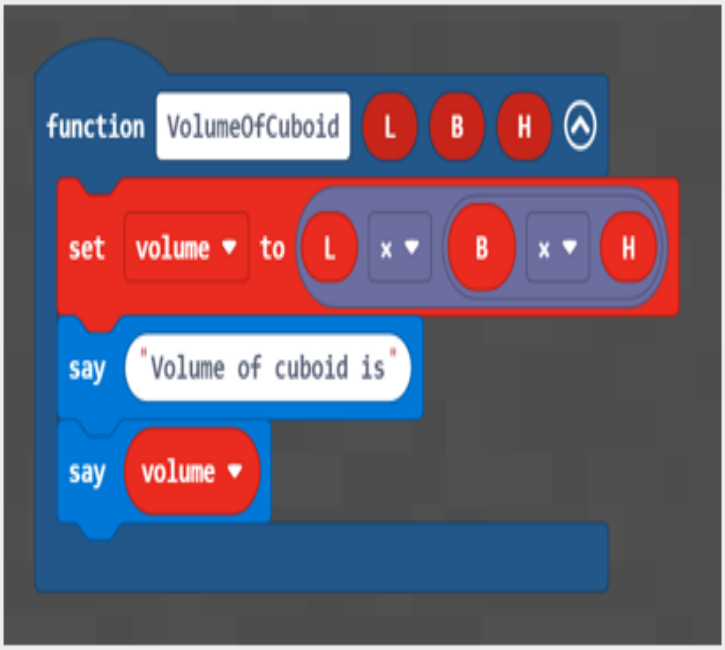
Step 2: Create a variable named **volume** and set it to 0. Place this block within the function.



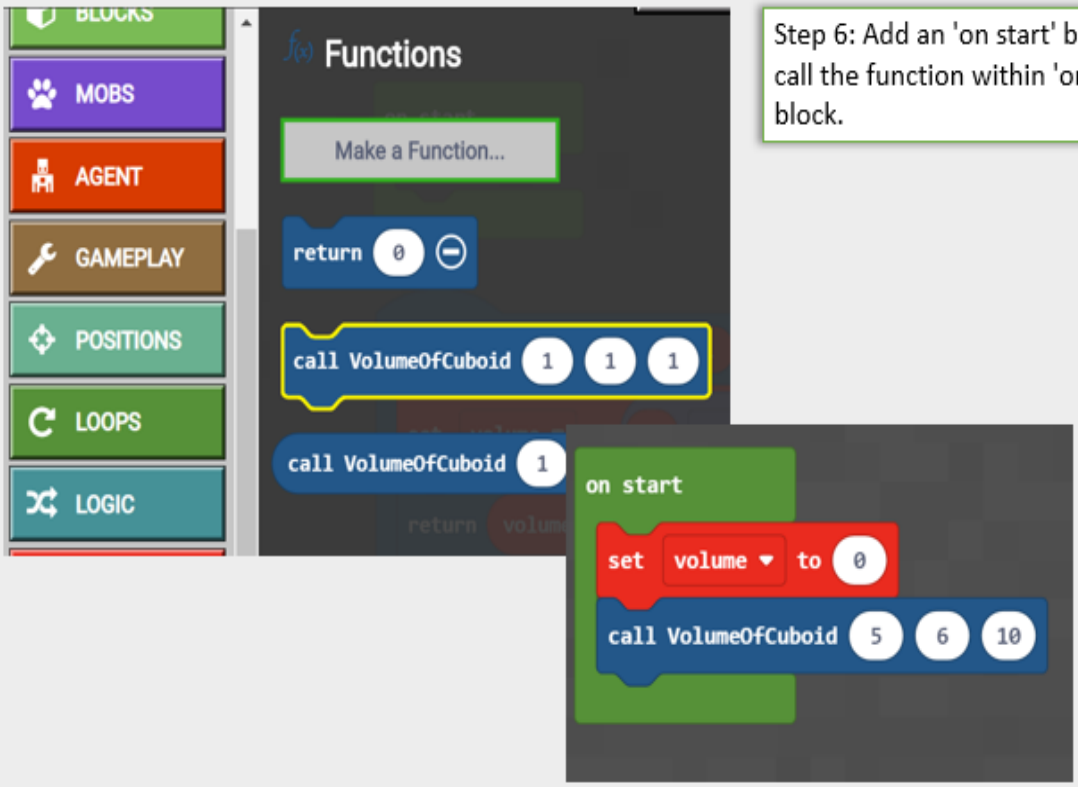
Step 3: Select multiplication block from math operators and place it inside the function.



Step 4: Add another multiplication block to multiply three numbers.



Step 5: Now add two 'Say' blocks to print the value of volume.



Step 6: Add an 'on start' block and call the function within 'on start' block.

Note: Minecraft is just one of the platforms to achieve this output. You can use many similar platforms available online to achieve similar output like – Scratch (https://scratch.mit.edu/) and Code.Org (https://code.org/)

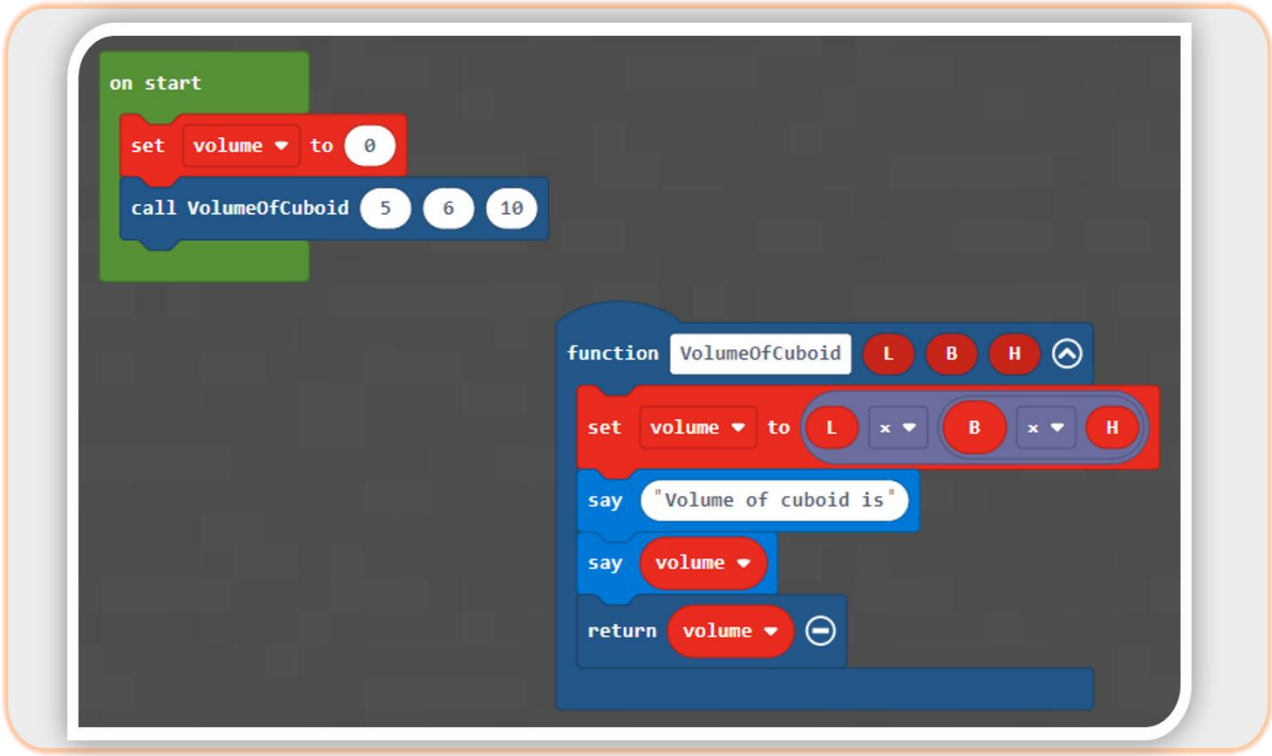
3.5 Returning value from a function

We have already used functions that takes arguments as inputs and does some operation with the arguments.

However, a function can also give an output or return a value. That makes a function more useful.

If we take the example of calculating the volume of a cuboid that we were working on, the VolumeOfCuboid did not return any value. When the program executes, the volume of cube gets calculated and assigned to the variable "volume". However, it becomes useless if the function doesn't return the variable "volume" to be used to assign the calculated volume. It's like in a restaurant, the waiter takes the order, goes into the kitchen and the food gets cooked but he/she never comes back with the food to the table. Returning a value is like the waiter coming back with the cooked food to serve to the table.

Let us continue with the example we have been working on in Minecraft. Click on the "Functions" button in the left panel and drag the "return" block to the function VolumeOfCuboid as shown below



3.6 Quiz time

Objective Type Questions

Question 1	How does the use of functions help in programming?
Option1	Reduces the repetition of the same set of statements many times.
Option2	Makes finding error in the code easier.
Option3	Enhances the logical transparency of the program
Option4	All the above

Question 2	Functions cannot return a value
Option1	True
Option2	False

Standard Questions

1. What are functions in programming?
2. What are the benefits of using functions?
3. Is it possible to return a value from function? If yes, how?
4. What are function parameters?

Higher Order Thinking Skills(HOTS)

1. Define a function to calculate the cube of a number. The function should take a number as argument and return the cube of that number.
2. Define a function to calculate volume of a sphere. The function should take radius of the sphere as argument. Use the function for calculating cube of a number to calculate the volume of the sphere. The function should return the calculated volume. Call the function to get the volume of a sphere.
3. Define a function to calculate the simple interest of an amount. The function should take the amount, the rate of interest and the time duration as arguments. The function should return the calculated simple interest.

Applied Project

1. **Problem Statement**: A branch manager of a reputed bank wants you to develop a software to calculate the simple interest. Create a program on Minecraft to calculate the simple interest using functions.
   Let the Principle amount = 100000
   Time =2 years
   Rate =5 %

3.7 What have you learnt in this chapter?

- Functions help us in increasing code-reusability
- Inputs to a function are called function parameters
- Functions can return value

PROGRAMMING WITH ARRAYS

4.1 What will you learn in this chapter?

- What are Arrays?
- What is sorting of an Array?
- What is searching in an Array?
- Apply the concept of Arrays in block coding and python using some fun examples

4.2 What are Arrays?

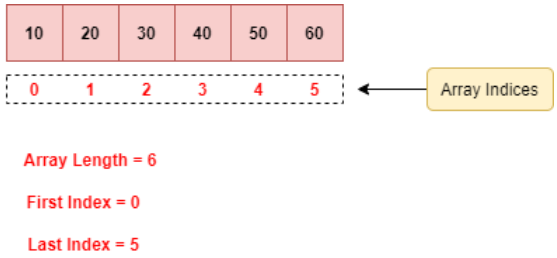
Arrays are collection of similar data type variables. Arrays do not support different data types in same collection. For example, you can make an Array of Integers as well as another Array of Strings. However, you cannot make an Array having Integer and Strings in same Collection.

In real world, you can consider books in a library to be the example of arrays where all the shelves have a common data type (read – book) in it.

Arrays improve readability of code by using a single variable for a large set of data. However, there are following limitations that we need to note while using Arrays.

- You can only store variables with homogenous data types in an Array
- Arrays are always static and fixed in length and size
- The variables in an Array are always ordered sequentially with index starting with 0

Consider an Array with variables of Integer Data Type stored in it. Below is the diagram that displays how data and indexes are structured in such an Array:



4.3 Activity: Playing with colors

Consider you have a list of items (a list of colors, for example), storing them in individual variables could look like this:

```
Color1 = "red"
Color2 = "green"
Color3 = "blue"
```

We will now see how to use a single variable to store all the colors

```
Color = ["red", "green", "blue"]
```

The above code snippet that we see is an example of array.

Arrays are special type of variables which can store a collection of values.

The values of an array can be accessed by referring to an index number.

For example, to get the first item in an array you will write your code like below

```
x = Color[0]
```

To modify the value of the first item in the array you will write:

```
Color[0] = "purple"
```

Question: How to create an array of a sequence of integers between 0 and 11.

Answer: Create an empty array. Begin a loop from 0 to 11. Append every item to the array.

```
m = []
for i in range(11)
  m.append(i)
print(m)
```

append() method is used to add an element to an array.

To remove an element of an array, use the pop() method.

```
colors = ["red", "green", "blue"]
colors.pop(1)
print(colors)
```

remove() method can also be used to remove an element from the array. However, remove() only removes the first occurrence of a specified value.

4.4 Sorting an Array

Consider the following array:

As you can see the items are not arranged in any order. Sorting is the process of ordering items in a collection. Above array when sorted in ascending order will look like below.

Python has inbuilt sort function to order an array. The sort() method sorts the list ascending by default.

Example:

```
numbers = [8,3,10,5]
numbers.sort()
print(numbers)
```

Result:

```
[3,5,8,10]
```

4.5 Searching in an Array

Python uses indexing as a method to search for an element in an array.

```
x = ["R", "D", "F", "H", "I"]
print(x.index("H"))
```

If you run the above code, you would get 3 as output.

The index method works on numeric arrays too:

```
x = [2,7,6,4,9,5,11]
print(x.index(9))
```

If you run the above code, you would get 4 as output.

4.6 Using Python to arrange an array in ascending order using bubble sort in MakeCode

Before we start this activity let us first understand what bubble sort is:

Bubble sort is a method of sorting that works by repeatedly swapping adjacent elements if they are in incorrect order. In this problem, you need to sort a given set of numbers in ascending order using bubble sort.

Let us consider a set of numbers 1, 5, 4, 3, 2; that needs to be sorted in ascending order. If the number on the left is greater than the number on the right, it swaps the two numbers. This is done until the numbers are sorted in ascending order.

- Following is the list of numbers to be sorted in ascending order:
- We compare the first two numbers and find out which one is greater.
- We start with 1 and 5. Since 5 is already greater than 1, no change is made

- Then we compare the numbers 5 and 4
- Since 5 is greater than 4, we will swap these two numbers.

- Next, we compare 5 and 3
- Since 5 is greater than 3, we will swap the two numbers.

- Lastly, we compare 5 and 2
- Since 5 is greater than 2, we will swap these two numbers

- The list of the numbers is rearranged as follows:
- Notice 5 is the largest number in this list and arranged at the last position.

- We again start from the beginning and compare 1 with 4. Since 4 is greater than 1, no change is made.

- Then we compare the numbers 4 and 3
- Since 4 is greater than 3, we will swap these two numbers.

- Next, we compare 4 and 2
- Since 4 is greater than 2, we will swap the two numbers.

- Lastly, we compare 4 and 5. Since, 5 is greater than 4, no change is made.

- We again start from the beginning and compare 1 with 3. Since 3 is greater than 1, no change is made.

- Then we compare the numbers 3 and 2
- Since 3 is greater than 2, we will swap these two numbers.

- Next, we compare the numbers 3 and 4. Since 4 is greater than 3, no change is made.

- We again start from the beginning and compare 1 with 2. Since 2 is greater than 1, no change is made.

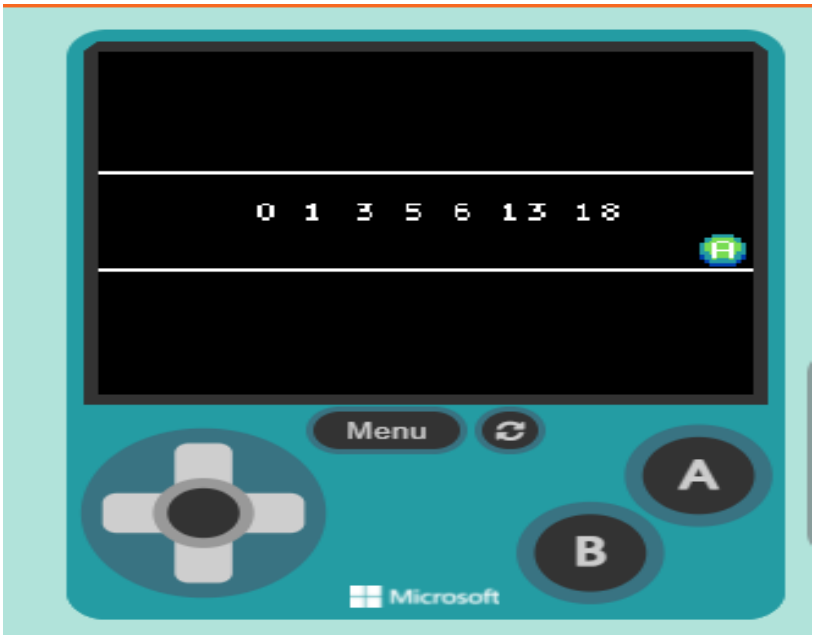
- The numbers are now all sorted in ascending order.

- The exercise of sorting is done until no more numbers need to be swapped.
- If we have 5 numbers, then the maximum time taken to sort the list will be 25 iterations.

Let us now write a python program to sort an array in MakeCode

Output for the code:

When you click on play the array will be sorted in ascending order.



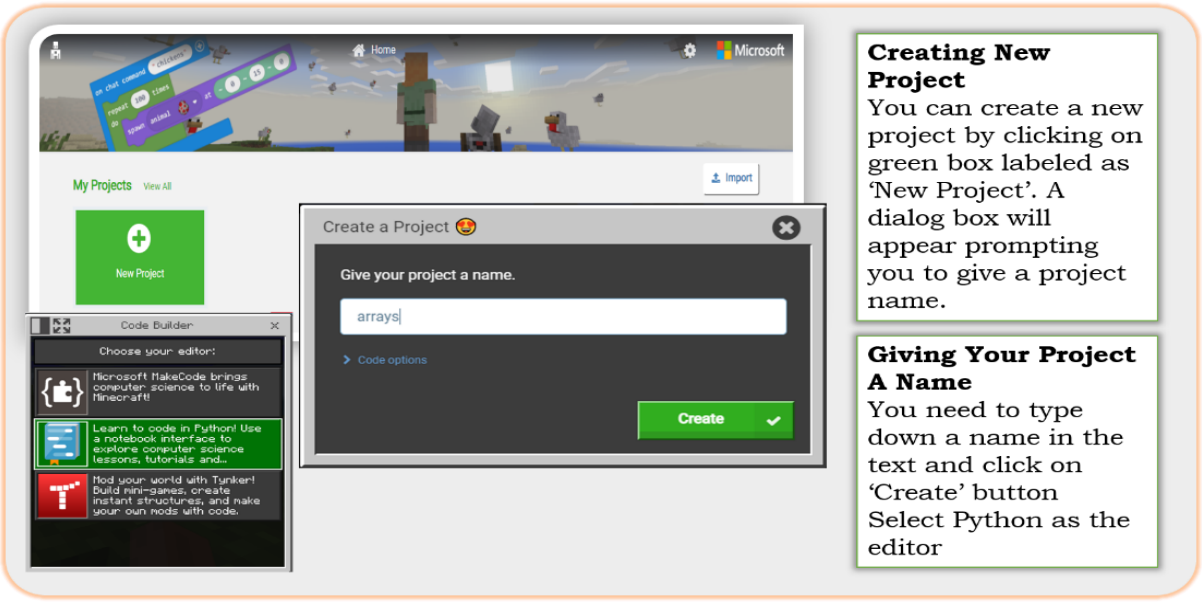
4.7 Activity: Fun in Minecraft using arrays

Let us now do some activity to understand Arrays.

At the end of this exercise, the final output on the screen should look like as shown in the below screenshot.

Final Output – Fun in Minecraft with arrays

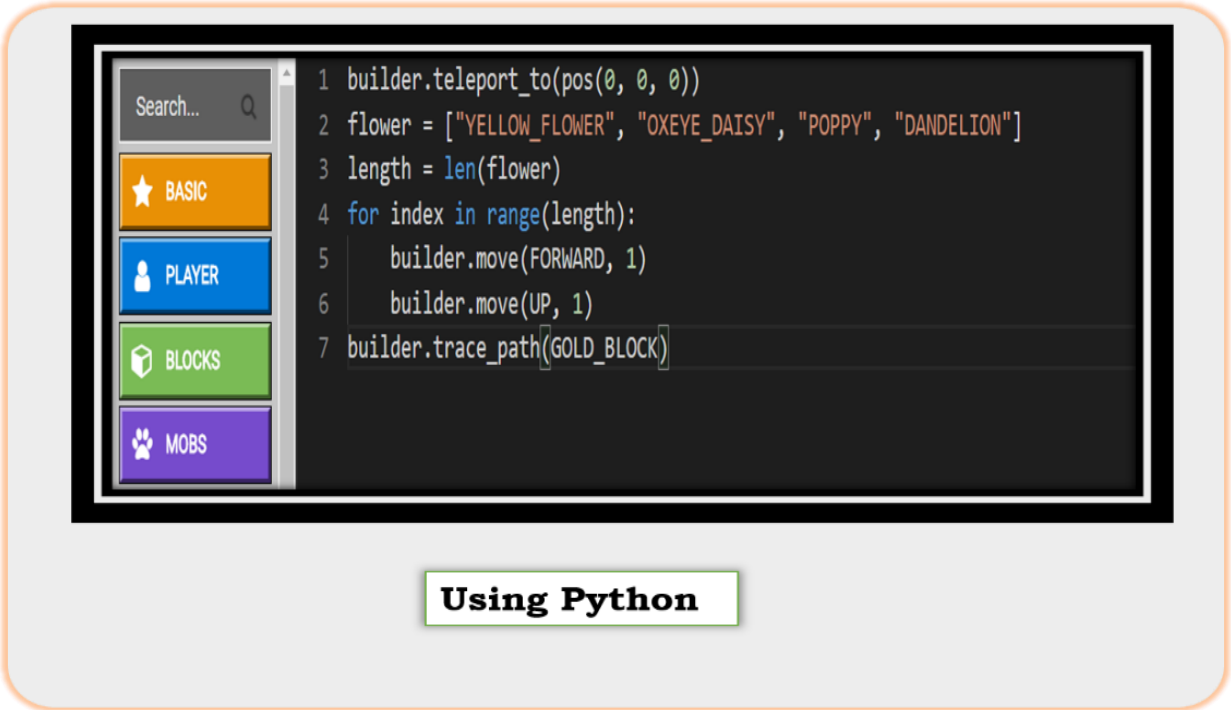
You should try this exercise on the Minecraft education edition.



Creating New Project
You can create a new project by clicking on green box labeled as 'New Project'. A dialog box will appear prompting you to give a project name.

Giving Your Project A Name
You need to type down a name in the text and click on 'Create' button
Select Python as the editor

In this activity we will create an array of flowers and calculate the length of the array to create a flight of stairs.



Here are the detailed steps:

Step 1: Teleport the builder to the entered position

Step 2: Make an array named flower

Step 3: Create a variable called length which is the length of the array flower

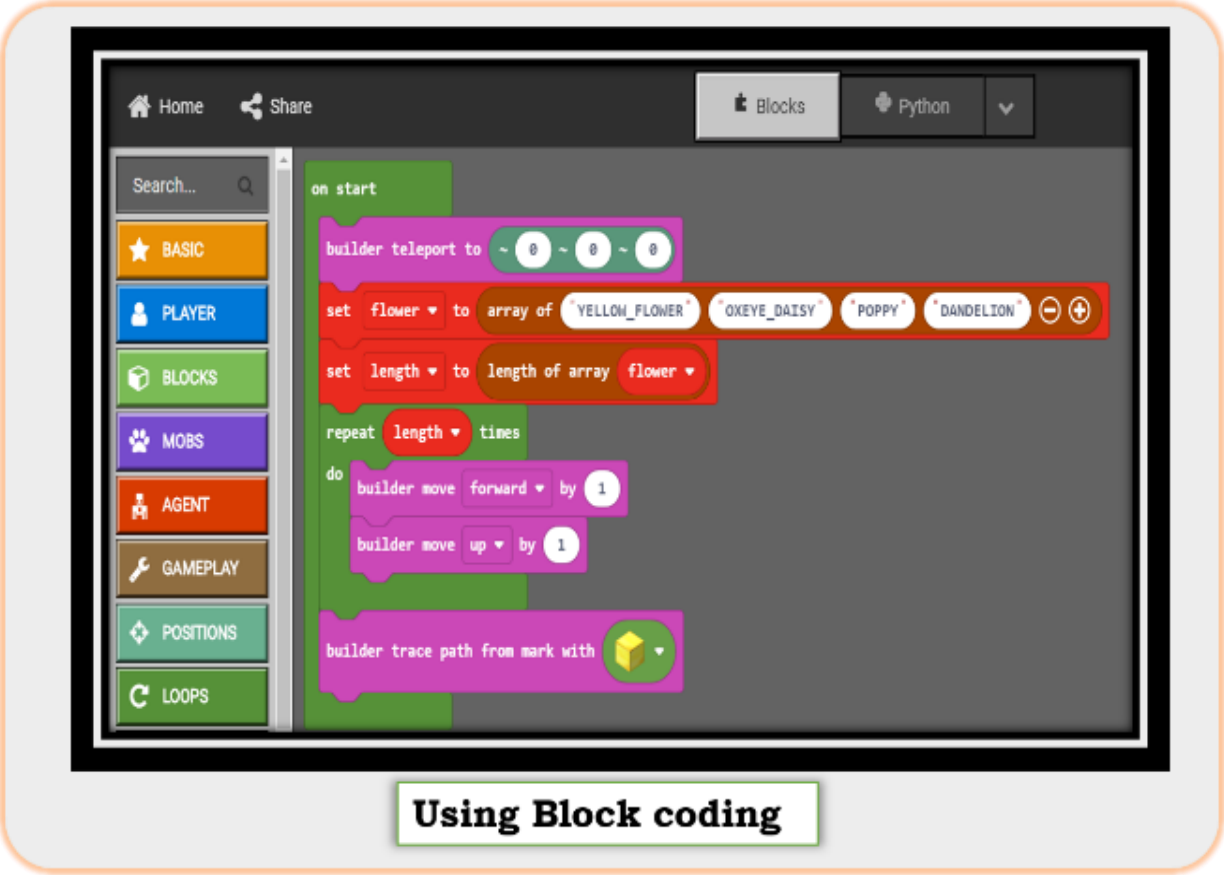
Step 4: Start a for loop which will run as many times as the length of the array

Step 5: Move the builder forward by 1 step and then up by 1 step

Step 6: Finally, the builder traces the path travelled and puts a block on the position

Step 7: When you hit play you will that there is a golden stair created.

Performing the activity using block codes:



Using Block coding

Note: Minecraft is just one of the platforms to achieve this output. You can use many similar platforms available online to achieve similar output like – Scratch (https://scratch.mit.edu/) and Code.Org (https://code.org/)

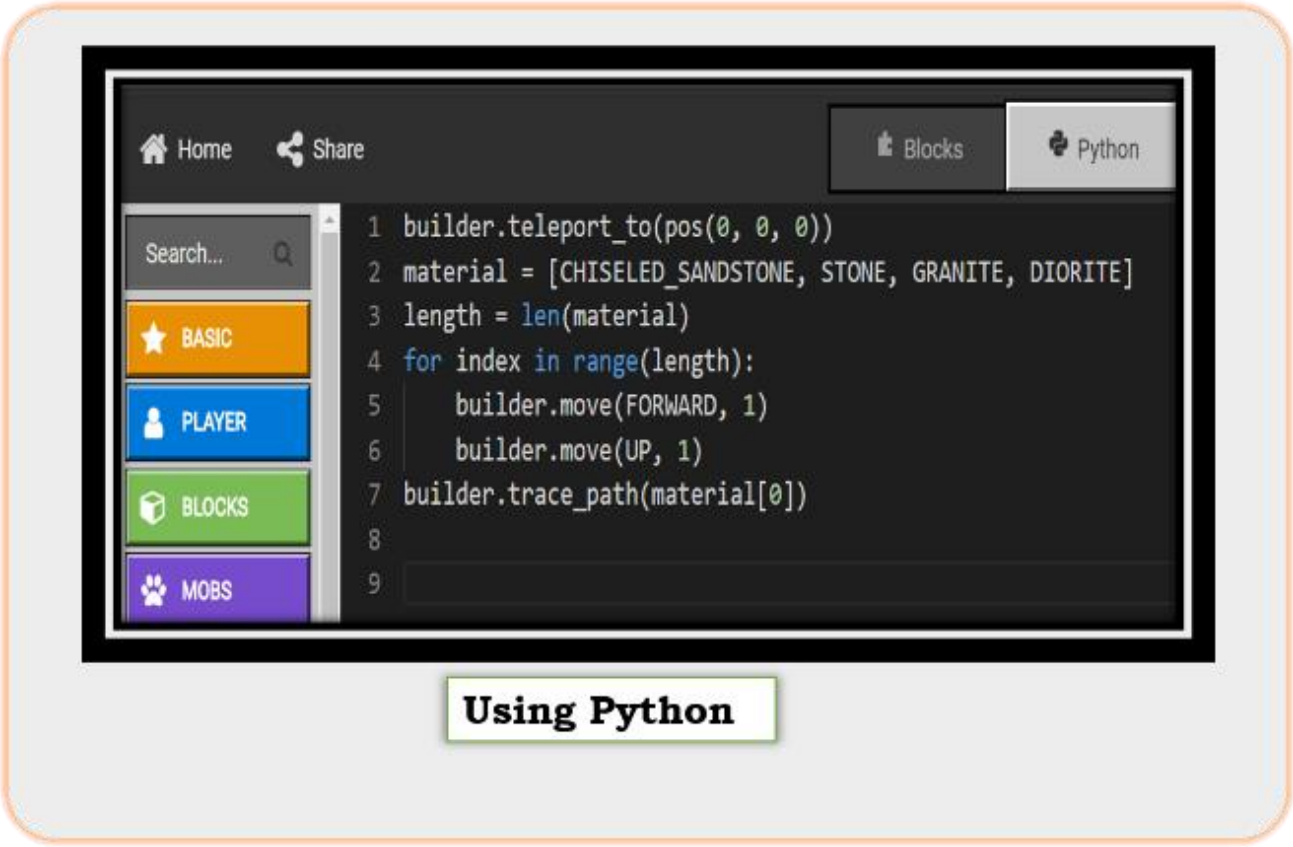
4.8 Activity: Building stairs

Now we are going to build a flight of stairs using the material present in the 1st position of the array.

At the end of this activity, the final output should look like shown in the image below:

Final Output – Building Stairs Activity

Using Python:



Here are the detailed steps:

Step 1: Teleporting the builder to the entered position

Step 2: Make an array named material

Step 3: Creating a variable length which is the length of the array

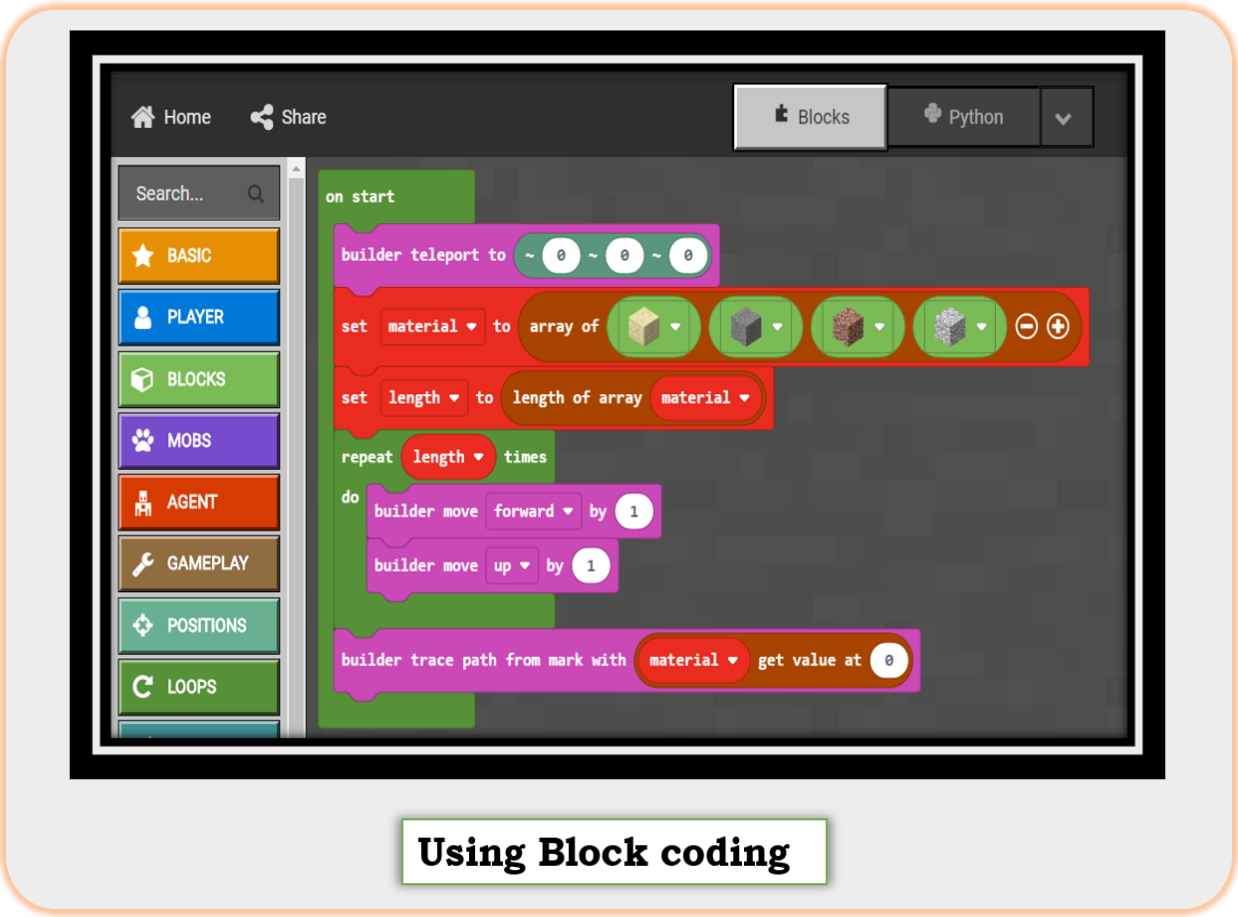
Step 4: Starting a for loop which will run as many times as the length of the array

Step 5: Moving the builder forward by 1 step and then up by 1 step

Step 6: Finally, the builder traces the path travelled and puts a block of material present on the 1st position (Chiseled Sandstone) in the array material on the traced path

Step 7: When you hit play you will that there is a Sandstone stair created.

Using Block code:



Using Block coding

Note: Minecraft is just one of the platforms to achieve this output. You can use many similar platforms available online to achieve similar output like – Scratch (https://scratch.mit.edu/) and Code.Org (https://code.org/)

4.9 Quiz time

Objective Type Questions

Question 1	Which statement from below best describes Arrays?
Option 1	A data structure that shows a hierarchical behavior
Option 2	Container of objects of similar data types
Option 3	Array is not a data structure

Question 2	Which of the following are advantages of arrays?
Option 1	Easier to store elements of similar data type
Option 2	Elements stored in an array cannot be sorted

Question 3	Which of the following are disadvantages of arrays?
Option 1	There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size
Option 2	Index value of an array can be negative
Option 3	Elements are sequentially accessed

Standard Questions

1. What are arrays in programming?
2. Explain how arrays are indexed in programming.
3. Explain how you can sort an array {67, 23, 98, 19} using Python?
4. How do you search a particular value from an array in Python?

Higher Order Thinking Skills(HOTS)

1. Make an array of animals (10 animals in the array) and use for loop which runs as many times as the length of the array to drop horses from the sky.
2. Make an array of animals (10 animals in the array) and use for loop which runs as many times as the length of the array to drop animal present on the 4th position (Tip: The 4th position will be indexed as 3) from the sky

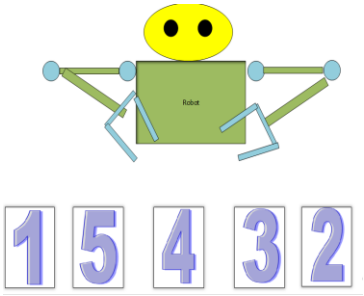
Applied Project

Problem Statement: There is a robot which can read the numbers written on the cards only when kept close to its eyes. It can pick only one card containing numbers in its left hand and another in its right hand. Can you help it to arrange the cards in ascending order using bubble sort? Let's try it on the numbers 1, 5, 4, 3, 2.

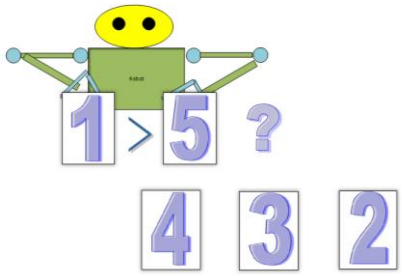
Tip: Arranging data in ordered sequence either in ascending or descending is called sorting. Using bubble sort each element is compared with its adjacent element until the desired result is reached.

Explanation:

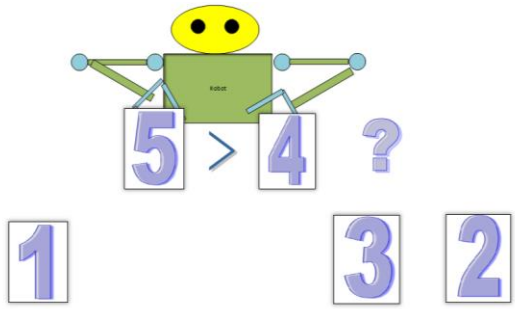
Let's take the numbers 1,5,4,3,2.

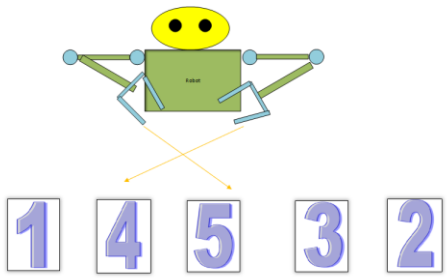


In the first move, the first two numbers 1 and 5 are compared. Since 1 < 5 therefore the numbers are not swapped.

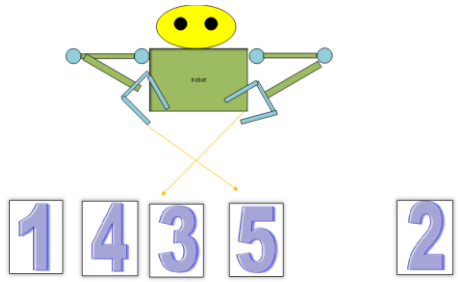


In the second move, the second and third numbers 5 and 4 are compared. Since 5 > 4 therefore the numbers are swapped.

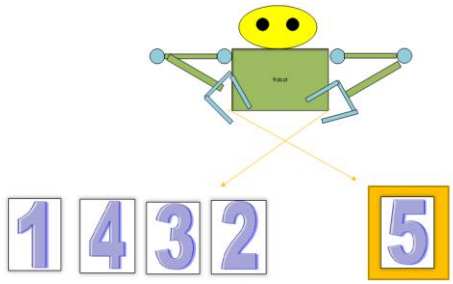




In the third move, the third and fourth numbers 5 and 3 are compared. Since 5 > 3 therefore the numbers are swapped.



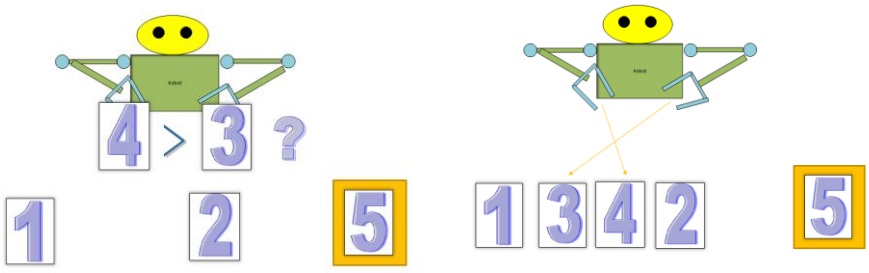
In the fourth move, the fourth and fifth numbers 5 and 2 are compared. Since 5 > 2 therefore the numbers are swapped.



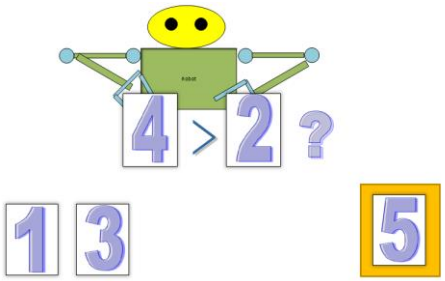
In the fifth move, the first and second numbers 1 and 4 are compared. Since 1 < 4 therefore the numbers are not swapped.

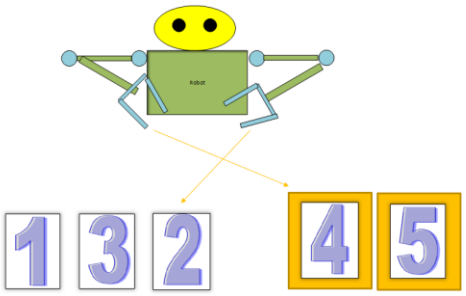
Note that the number 5 is sorted. Let us repeat the same steps for numbers 1 ,4,3 and 2.

In the sixth move, the second and third numbers 4 and 3 are compared. Since 4 > 3 therefore the numbers are swapped.



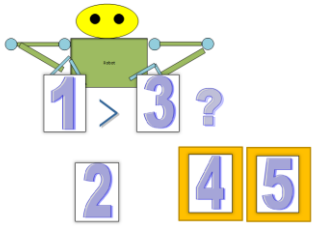
In the seventh move, the third and fourth numbers 4 and 2 are compared. Since 4 > 2 therefore the numbers are swapped.



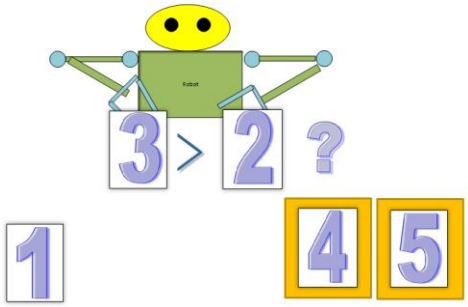


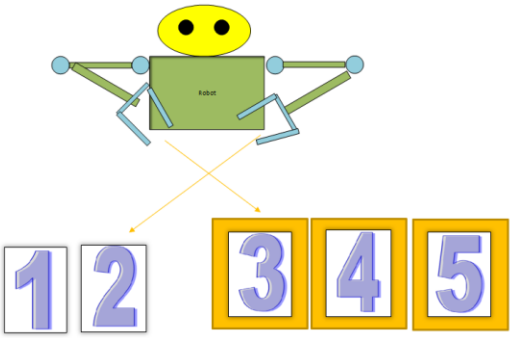
Note: The number 4 is sorted. Let us repeat the same steps for numbers 1 3 and 2.

In the eighth move, the first and second numbers 1 and 3 are compared. Since 1 < 3 therefore the numbers are not swapped.



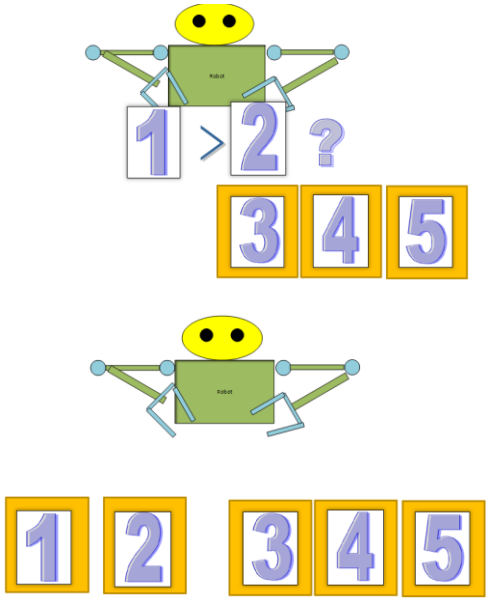
In the ninth move, the second and third numbers 3 and 3 are compared. Since 3 > 2 therefore the numbers are swapped.





Note: The number 3 is sorted. Let us now try for 1 and 2

In the tenth move, the first and second numbers 1 and 2 are compared. Since 1 < 2 therefore the numbers are not swapped.



After tenth moves we can sort the numbers in ascending order. This process of arranging numbers is called bubble sort.

4.10 What have you learnt in this chapter?

- We have understood the concept of arrays
- We have also learnt how to search a value in an array and how to sort an array
- We have also applied the concepts of arrays in block coding and python using some fun examples

Chapter 5

ADVANCED SEQUENCING

5.1 What will you learn in this chapter?

- What is a sequence?
- Why is sequencing important in programming?
- Sequencing with Loops and Conditions
- Using sequencing in Minecraft to do some fun activities

5.2 What is a Sequence?

A series of actions performed in a specific order is called sequence. For example, our daily routine. Our daily routine is a series of actions that we do every day as follows.

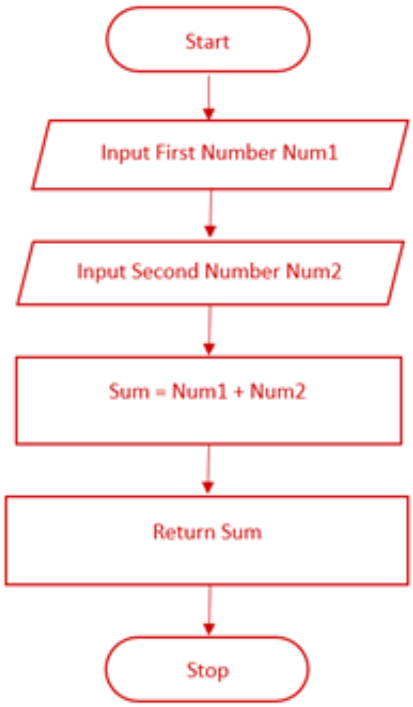
1. Wake up.
2. Have breakfast.
3. Take a shower.
4. Go to school.
5. Attend classes.
6. Have lunch.
7. Attend classes.
8. Get back home.
9. Do homework.
10. Watch TV.
11. Have dinner.
12. Go to sleep.

However, this routine might vary from person to person but for a specific person this might be the routine.

An **algorithm** is a set of steps to solve a problem. A program is written based on an algorithm to get the required result.

Sequencing in algorithms is arranging the steps of an algorithm in a correct sequence so that we get the correct outcome. Sequencing is important in algorithms. If the steps of an algorithm are not in correct sequence, we will not get the required result.

For example, if we make a simple algorithm of calculating the sum of two numbers, the steps would look like as the following in a flowchart.



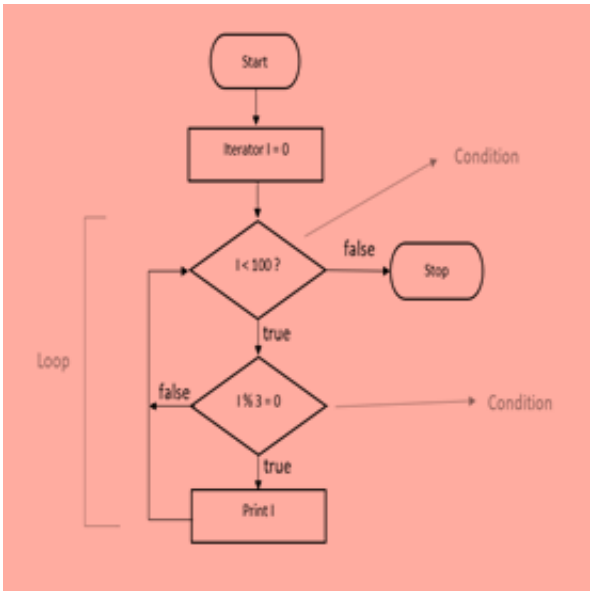
If we swap 2 steps and add Num1 and Num2 before taking Num1 as input, the program will not give the required result or give an error. This is called a **bug.**

5.3 Sequencing with Loops and Conditions

Most programs consist of 3 main components – Sequence, Selections (Conditions) and Loops.

Let's write a program to get all the numbers divisible by 3 between 0 and 100. The flowchart for the algorithm would be as follows.

In this example, we see that this program has a sequence of conditions and loops.



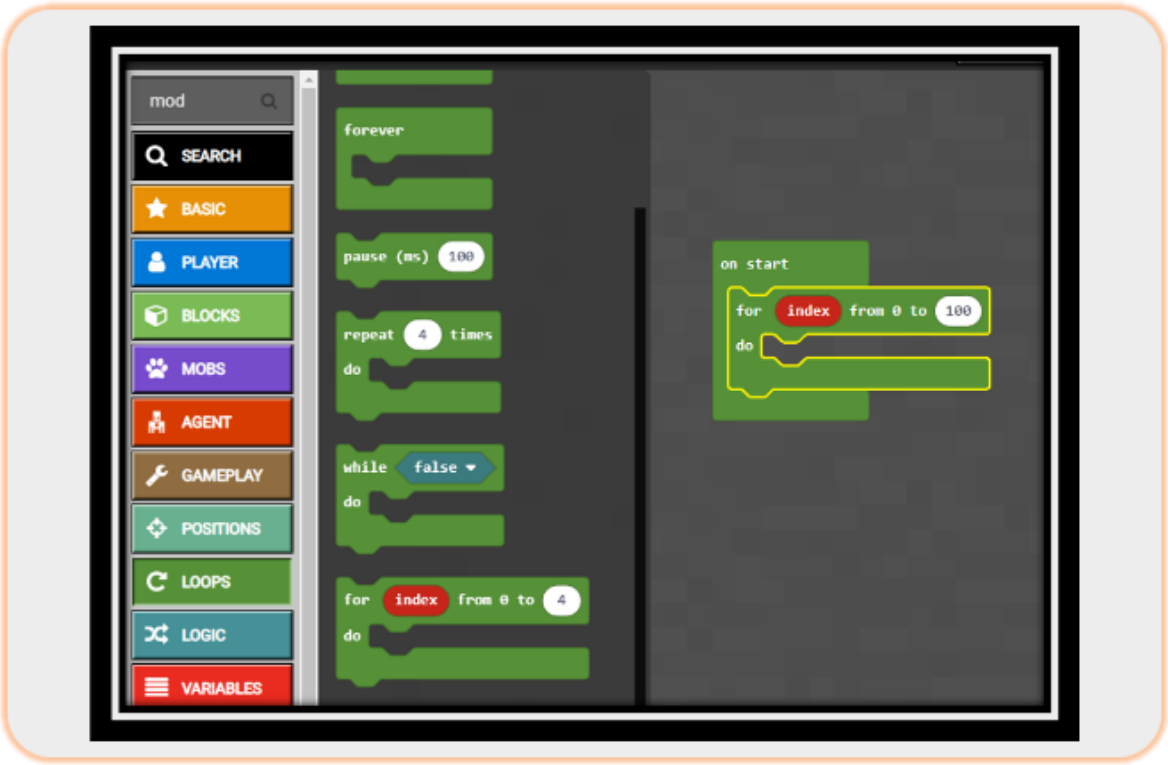
5.4 Activity: Print number divisible by 3 from 1 to 100

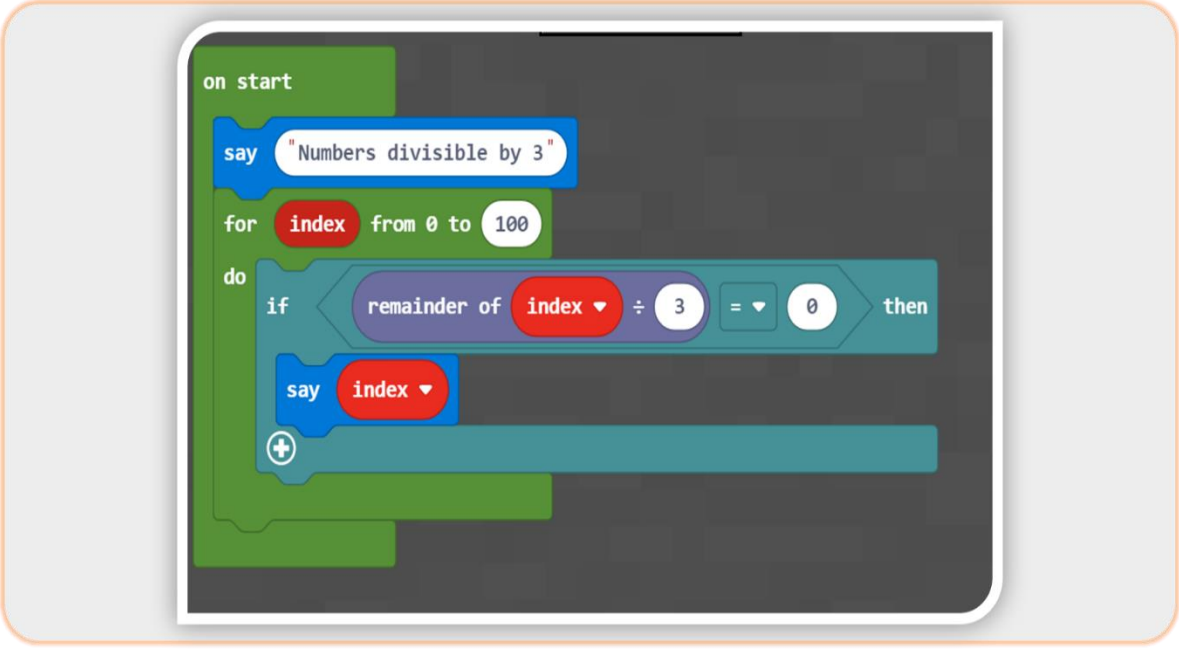
Let's try this activity in Minecraft using block code.

At the end of this activity the final output should look like shown in the image below:

Final Output – Print number divisible by 3 from 1 to 100 Activity

Let us start to replicate this activity to Iterate a loop from 0 to 100 as shown below





Note: Minecraft is just one of the platforms to achieve this output. You can use many similar platforms available online to achieve similar output like – Scratch (https://scratch.mit.edu/) and Code.Org (https://code.org/)

5.5 Quiz time

Objective Type Questions

Question 1	Step by step process of solving a problem is called?
Option 1	Algorithm
Option 2	Sequencing
Option 3	Loops
Option 4	None of the above

Question 1	A series of actions done in a specific order is called sequencing
Option 1	True
Option 2	False

Standard Questions

1. Explain what is a sequence in programming.
2. Draw a flowchart to demonstrate sequence of activities that you do while getting ready for school.
3. Write a pseudocode to explain sequence of activities that you perform while giving an exam.

Higher Order Thinking Skills(HOTS)

1. Write a program in block code to print all the prime numbers between 0 and 50
2. Write a program in block code to print all the multiples of 7 from 0 to 100

Applied Project

1. **Problem Statement**: Write a flow chart for the algorithm to determine if a number is perfect square.

5.6 What have you learnt in this chapter?

- We have understood the concept of advanced sequencing
- Significance of sequencing in programming
- We have also learnt sequencing in block coding through some fun examples

REFERENCES

Microsoft. 2021. Microsoft MakeCode Arcade. [Online]. [25 February 2021]. Available from: https://arcade.makecode.com

Microsoft. 2021. Microsoft MakeCode Arcade for Minecraft. [Online]. [25 February 2021]. Available from: https://arcade.makecode.com

Microsoft. 2021. Computer Science Subject Kit | Minecraft: Education Edition. [Online]. [25 February 2021]. Available from: https://education.minecraft.net/class-resources/computer-science-subject-kit

Microsoft. 2021. Minecraft MakeCode | Rain cats and dogs. [Online]. [25 February 2021]. Available from: https://minecraft.makecode.com/examples/cats-vs-dogs

Code of Ethics. 2021. Code of Ethics. [ONLINE] Available at: https://www.acm.org/code-of-ethics