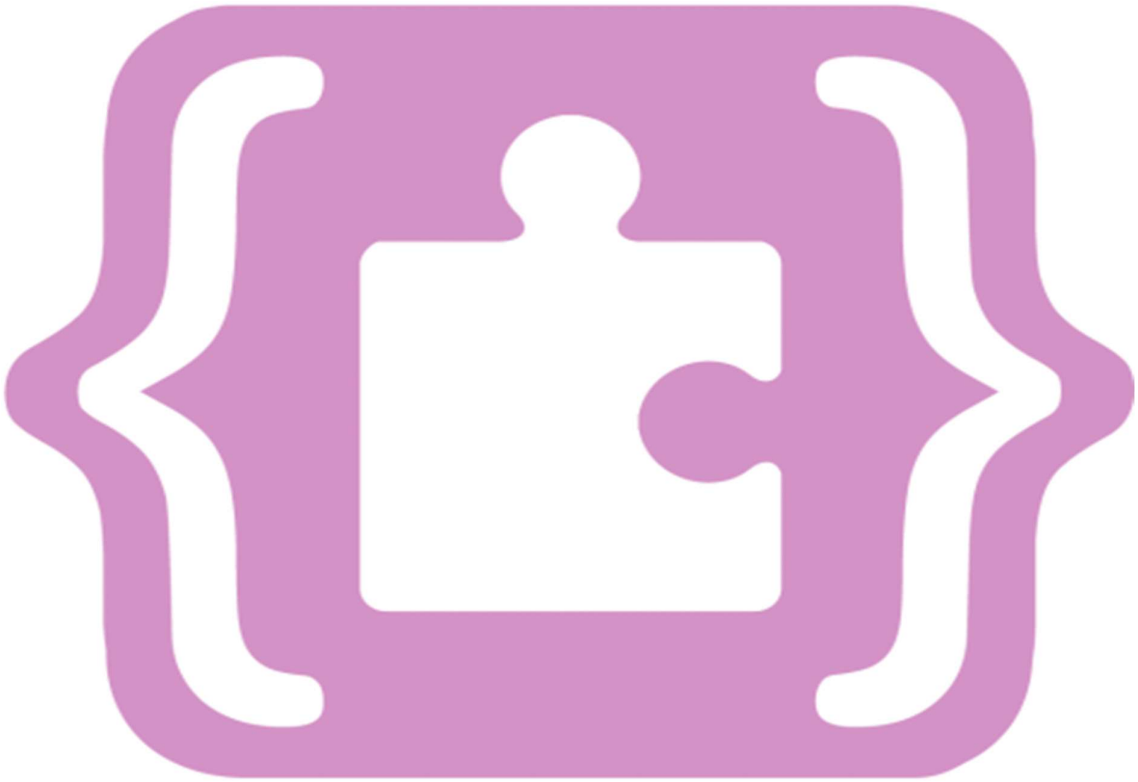
MakeCode Blocks



Compiled and Created By –
Harpreet Kour & Satnam Singh

This Manual created for education purpose; Content recreated from course available in public domain and Makecode and Minecraft

https://makecode.com/

What is Block

In 1975, Seymour Papert of the MIT Media Lab created a beginners' programming language called **LOGO**. He developed it based on research that showed that playing with blocks of code was a particularly effective way to teach programming concepts. Papert coined the term "constructionism" to describe the way that learners construct new knowledge by building on established knowledge. The blocks in MakeCode, and the blocks in Minecraft, are themselves models for the way that new learning happens through the application of concepts in an openended learning environment. Block-based programming languages such as Scratch and MakeCode build on Papert's research and are a great way for students to start learning about coding concepts without having to worry about syntax.

As opposed to text-based programming, block-based programming refers to programming language and IDE that separates executable actions into modular portions called blocks. These blocks are generally represented with icons that can be clicked and dragged to reorder them. Editable fields, like drop-down menus, allow users to provide further input. This graphical representation of the code can demonstrate the process to new users who may be unfamiliar with programming.

Data Types

A *type* refers to a class of data and the operations permitted on that class of data. The following built-in types are supported for the Arcade:

Basic (primitive) types

- Number: an integer number (32-bit signed)
- String: a sequence of characters
- Boolean: true or false

Complex types

Array: a list of items of a primitive type

Functions

Function: code you can reuse anywhere in a program

Images and Sprites

Image: rows and columns of color pixels that make a picture

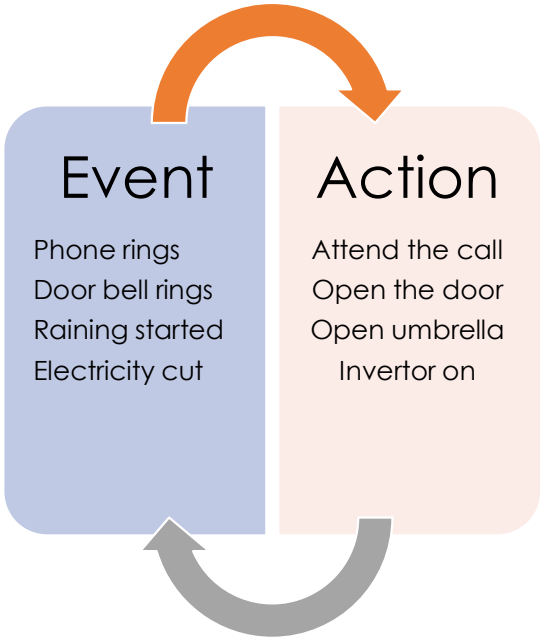
Sprite: an operation object to locate and move an image

User data

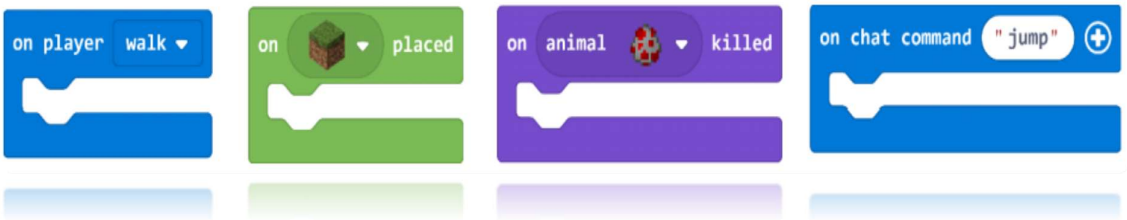
TypeScript allows you to create user-defined classes of data.

Events

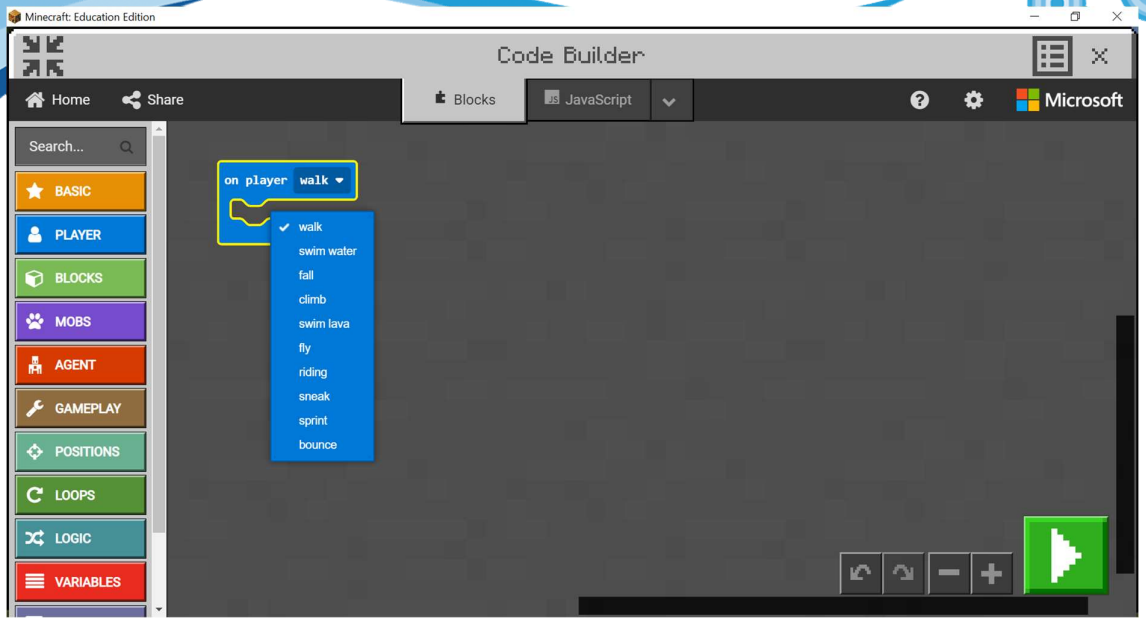
An "event" in computer science is an action or occurrence that is detected by a computer. For example, when someone clicks the button on their mouse, it generates a "mouse click event" for the computer. In real life, there are also events that may be associated with a following action, like Cause-and-Effect.



In programming, an event handler is a part of your program that runs when specific events happen (it "handles" the event). In MakeCode, these event handler blocks look like a square with a gap in the middle, and usually start with the word "on"

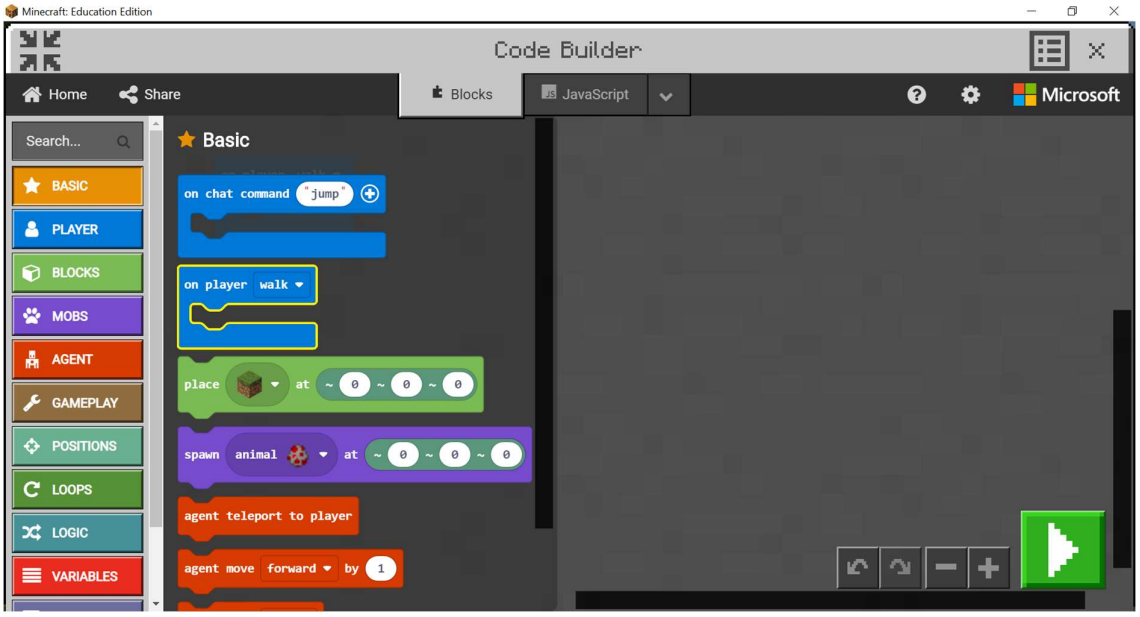


The "On player walk" block is an event handler that looks for a specific action by the player. Its pull-down menu lists all sorts of actions a player might perform at any given time, such as walking, jumping, attempting to swim in lava, and more.



You can configure this event handler to cause something to happen when a player is walking. For example, you can leave flowers everywhere you walk!

Steps: 1. From the Player Toolbox drawer, drag the On player walk block into the coding Workspace



From the Blocks Toolbox drawer, drag the Place at block under the On player walk block until you hear and see it snap into place.