

June 2020



Programming 2.0

Introduction – Week 1

Guru Anginthayya

Agenda – Week 1

- + Course Introduction
- + Computational thinking
- + Programming
- + Micro:bit introduction
- + Hello World project
- + Computer – Common Features
- + Review



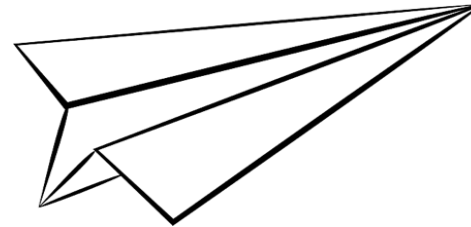
Course Introduction – draft

- + Week 1 - Computing fundamentals
- + Week 2 - Data handling
- + Week 3 - Musical micro:bit
 - + Exploring sound & music
- + Week 4 - Exploring micro:bit
 - + Fun projects
- + Week 5 - Cybersecurity
- + Week 6 - Learning review & Project work
- + Logistics – email and homework



Paper Airplane Challenge

- + Breaking down a complex problem into smaller parts.
 - + Fold the paper
 - + Make the wings
 - + Make it more attractive
- + Focusing on the important parts of a system.
 - + Not focusing on air circulation, brightness etc

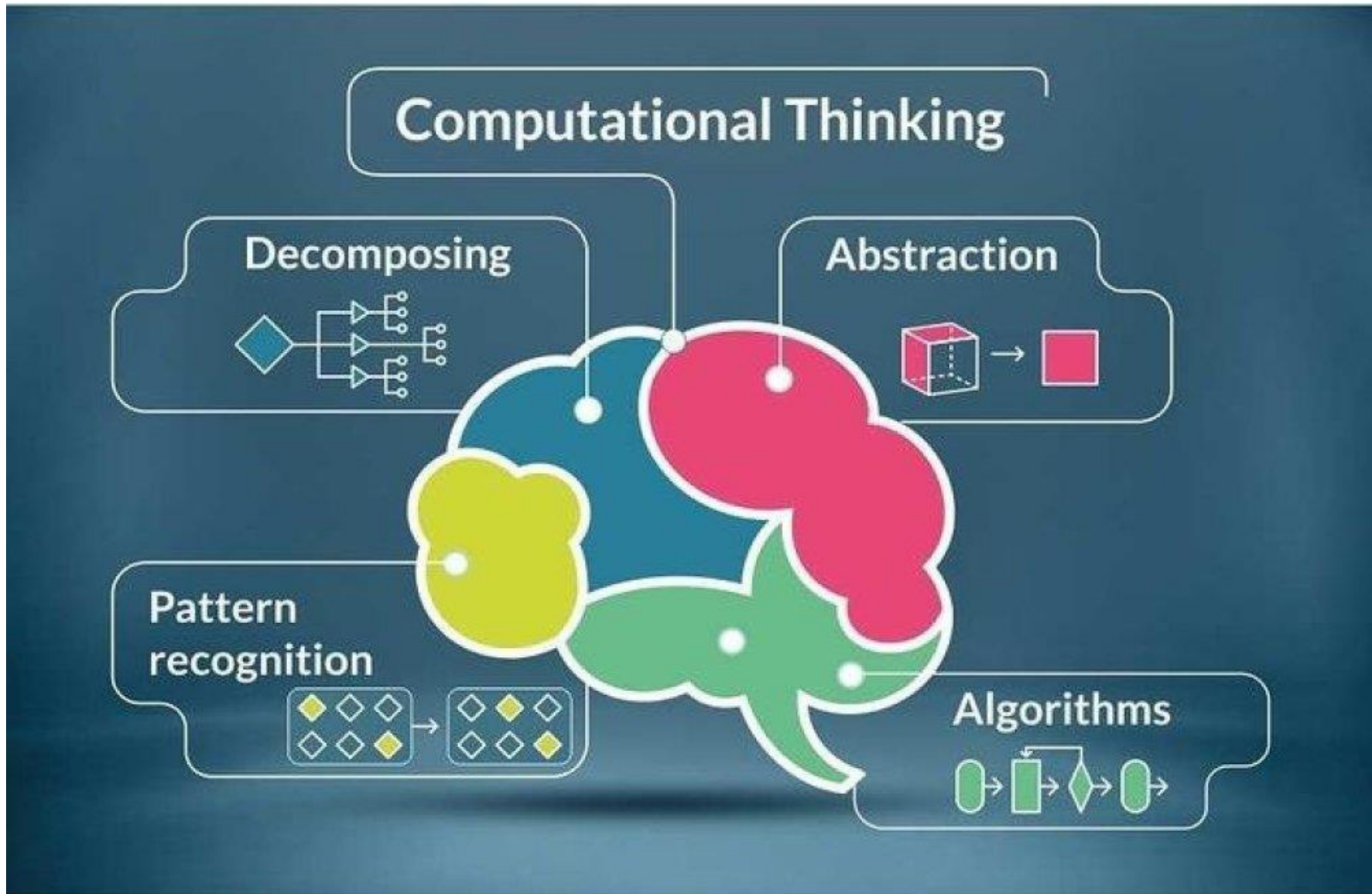


Paper Airplane Challenge

- + Looking for similarities.
 - + what planes usually look like (small, big planes)
 - + You identified common features
 - + wings, nose, windows
- + The steps or rules to complete a task
 - + The instructions you created for someone to follow to create a giant paper airplane is an algorithm
- + Testing/ Evaluation
 - + Checking your plane flies

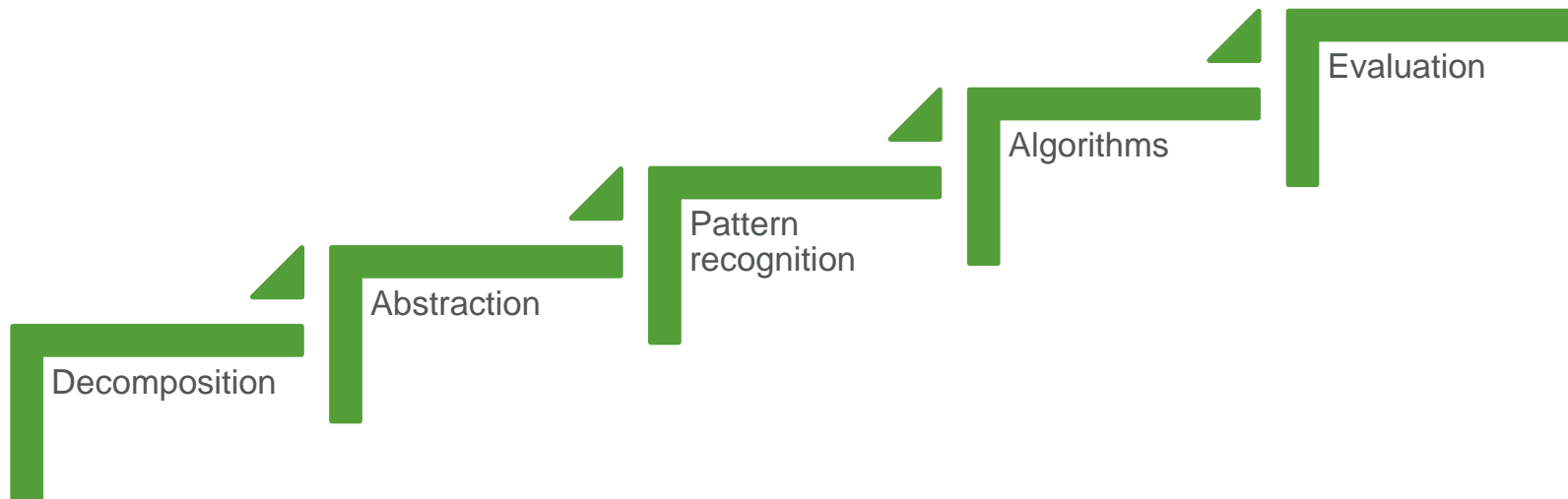


Fundamentals of Computational thinking



Fundamentals of Computational thinking

- + Computational thinking - Considering a problem in a way that a computer can help us to solve it.
 - + Even before writing the instructions for the computer we need to
 - + Understand what the problem is
 - + Consider possible solutions
 - + Use a computer to help solve the problem



Fundamentals of Computational thinking

- + **Decomposition** - breaking down a complex problem or system into smaller, more manageable parts
- + **Abstraction** – focusing on the important information only, ignoring irrelevant detail
- + **Pattern recognition** – looking for similarities among and within problems
- + **Algorithms** - developing a step-by-step solution to the problem, or the rules to follow to solve the problem
- + **Evaluation** - Evaluation means considering if a solution is 'fit for purpose'.
 - + Is it:
 - + suitable for the user?
 - + easily understood?
 - + efficient?
 - + the best that can be done with the resources available?



Fundamentals of Programming

+ Programming

- + The process of writing computer code to create a program.

- + Translating the steps in an algorithm into a language the computer can understand so the problem can be solved.

+ Programming languages

- + A language we can use to write the steps of the algorithm so the computer can understand and run it.

- + What programming languages have you already used?

- + Write program

- + Test/Debug



Fundamentals of Programming

+ Programming

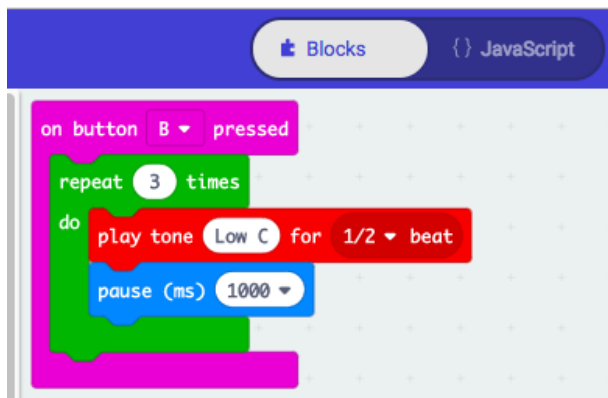
- + The process of writing computer code to create a program.
- + Translating the steps in an algorithm into a language the computer can understand so the problem can be solved.

+ Programming languages

- + A language we can use to write the steps of the algorithm so the computer can understand and run it.
- + What programming languages have you already used?

+ Write program

- + Graphical vs Text based



A screenshot of a block-based programming environment. The interface shows a blue header with 'Blocks' and 'JavaScript' tabs. The main workspace contains a pink 'on button B pressed' block, a green 'repeat 3 times' block, a red 'do' block containing a 'play tone Low C for 1/2 beat' block, and a blue 'pause (ms) 1000' block.



```
1 input.onButtonPressed(Button.B, function () {
2   for (let i = 0; i < 3; i++) {
3     music.playTone(131, music.beat(BeatFraction.Half))
4     basic.pause(1000)
5   }
6 })
7
```



Programming Language Features

- + Iteration - Repeating steps in an algorithm or program. Also called repetition and loops.
 - + Makes algorithms and programs quicker and simpler to write.
 - + Removes unnecessary steps
 - + Makes programs more efficient to run

INPUT when button B p

REPEAT 3 times

OUTPUT Low C for 1/2

Pause for 1 second



Programming Language Features

- + Selection - A decision or a question that tells a program to look at the answer, then follow a particular sequence
 - + allows choices/decisions to be included
 - + enables different pathways through an algorithm and program to be included

REPEAT continuously

INPUT **IF** button B pressed

REPEAT 3 times

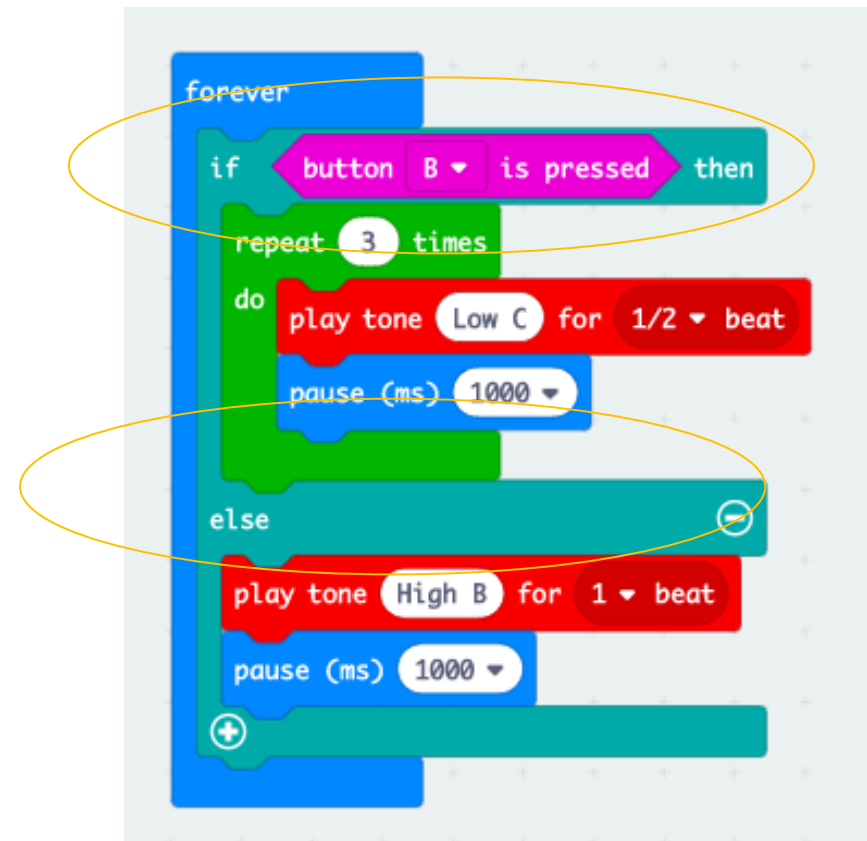
OUTPUT Low C for ½ beat

Pause for 1 second

ELSE

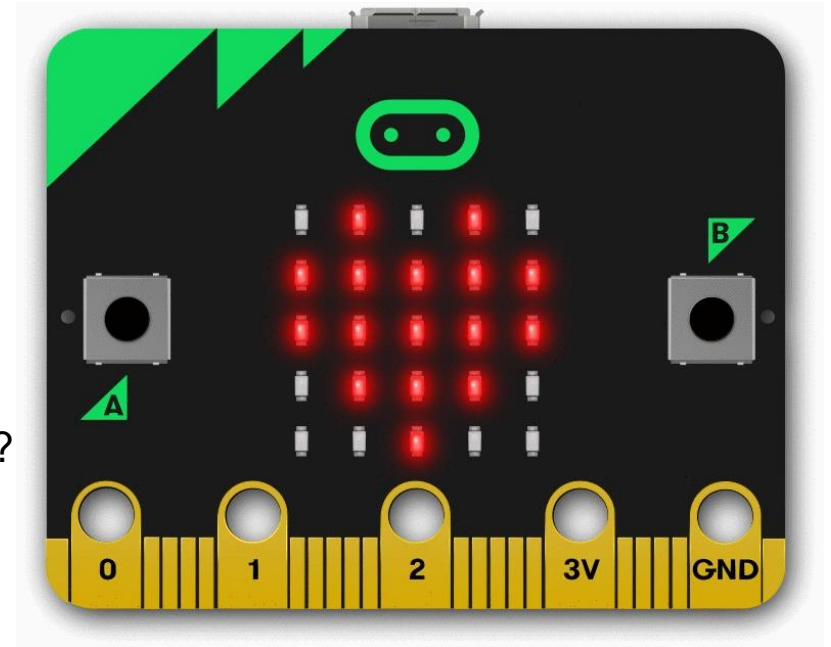
Play High B for 1 beat

Pause for 1 second

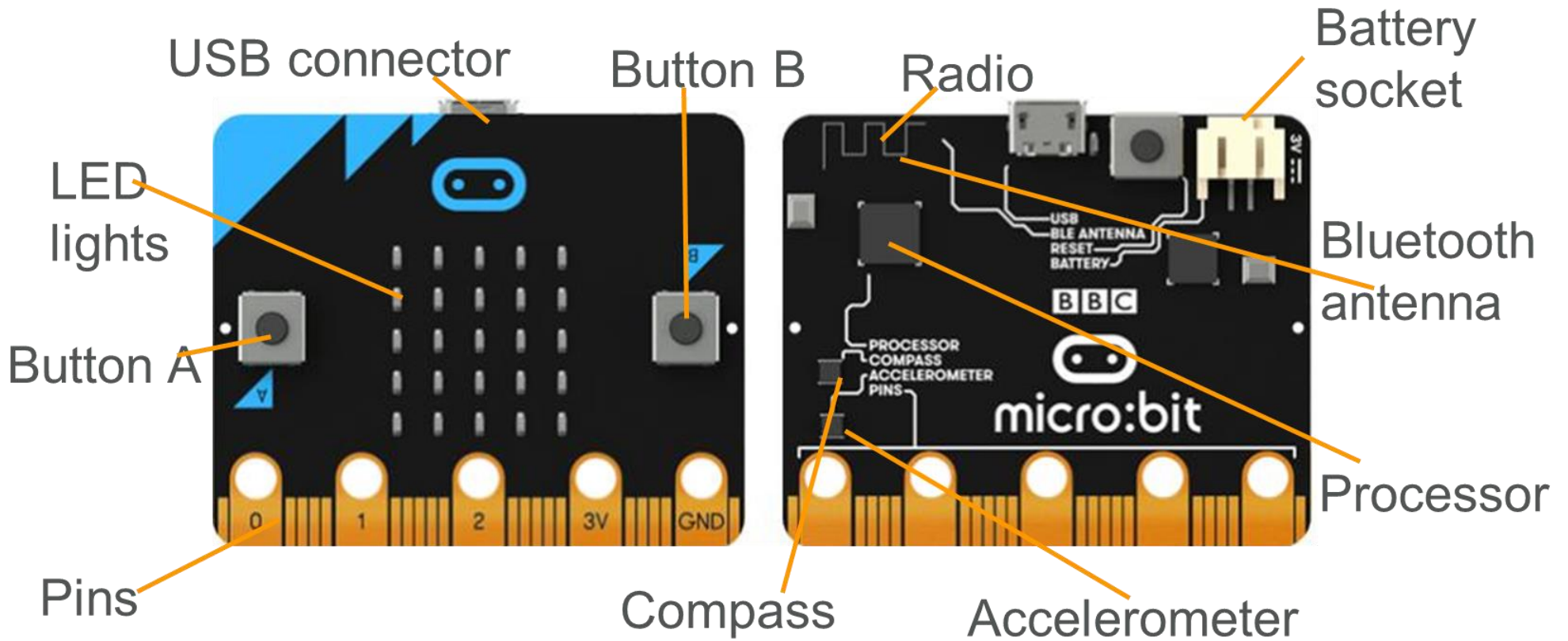


Micro:bit

- + What is it?
 - + Tiny computer – Smaller than Credit card (40 mm * 50 mm)
 - + Processor – (16 mHz CPU)
 - + RAM – (16 KB RAM + 256 KB Flash memory)
 - + Battery or USB as power cable
- + Homework – to find out the configuration of the laptop?
- + Program – HEX file
- +



Micro:bit – Parts



Micro:bit – Hello World program!

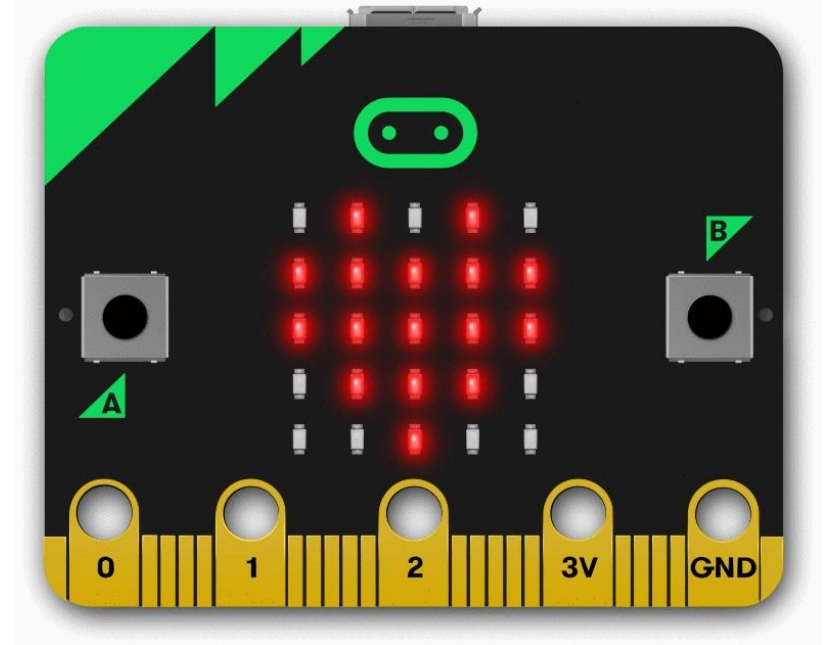
+ <https://makecode.microbit.org/>

+ Download

+ Download hex file

+ Download via WebUSB

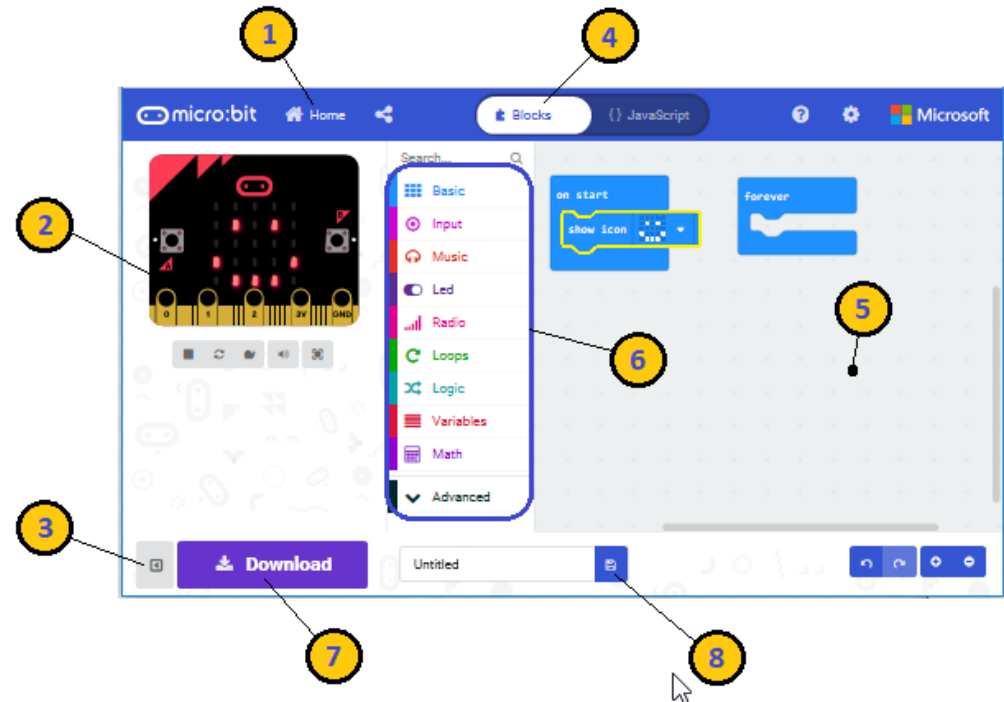
+ iOS app - Microsoft MakeCode (bluetooth transfer)



Microsoft MakeCode

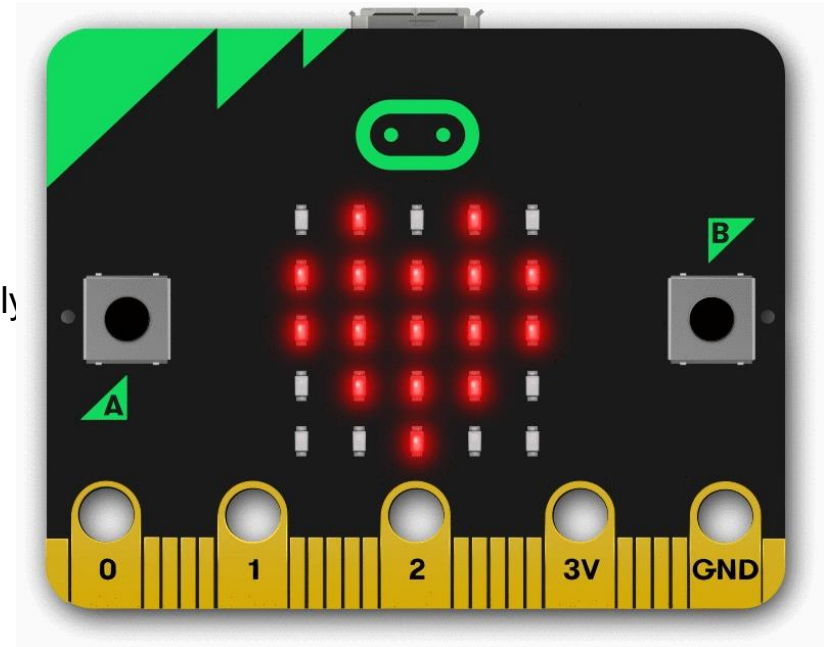
+ <https://makecode.microbit.org/>

1. **Home Screen** to start a new project or open an existing project
2. **Simulator** shows what your program will look like when running on a micro:bit
3. **Hide** or **Show** the simulator pane
4. **Blocks** or **JavaScript**
5. Programming **Workspace**
6. Blocks **Toolbox**
7. **Download** program to the micro:bit
8. **Save** the project



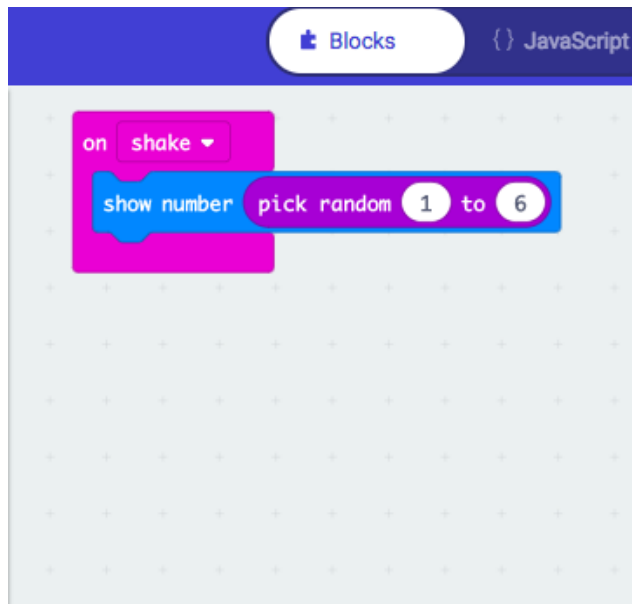
Computer – Common Features

- + Hardware - The physical parts of a computer system
 - + LED – I can display text, images and numbers.
 - + Compass - I detect the earth's magnetic field
 - + Radio & Bluetooth - I can be used to communicate wirelessly
 - + USB - I can be used to connect micro:bit to a computer
- + Common features of computer systems
- + Input and output devices, hardware and software
- + Programming micro:bit
- +



Computer – Software

- + Software – The programs that run on a computer
 - + showing random numbers in dice
 - +

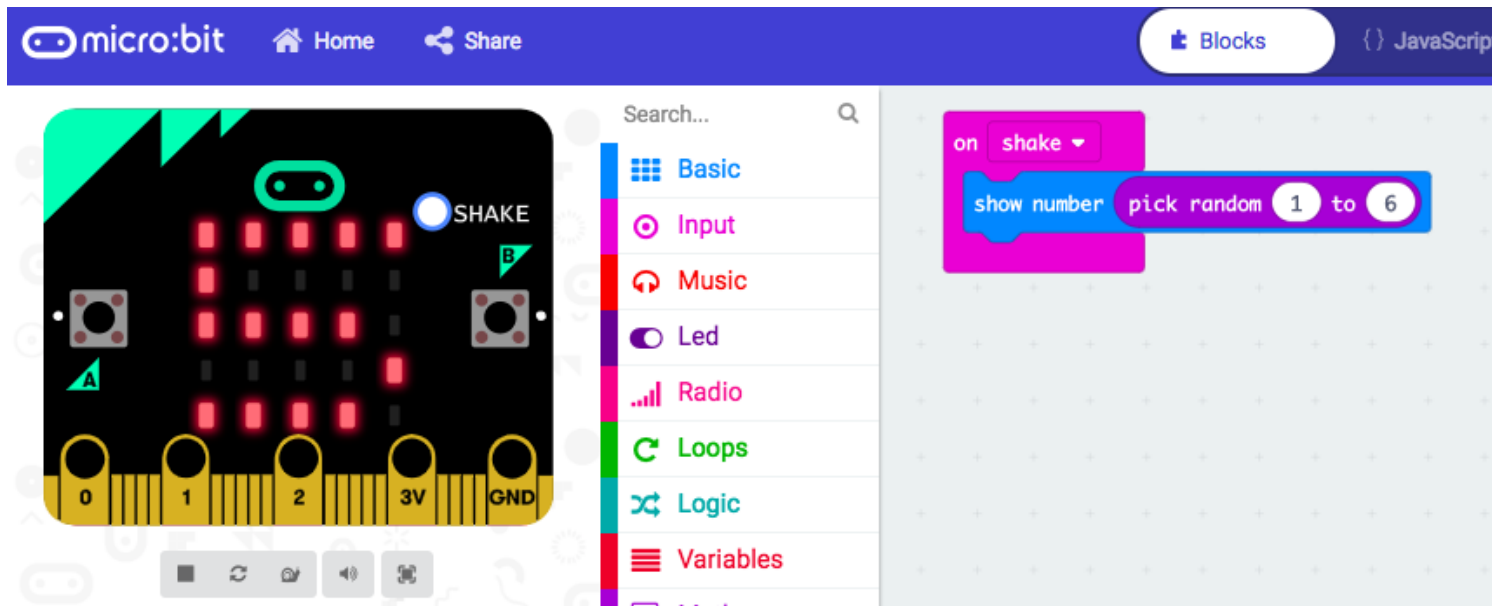
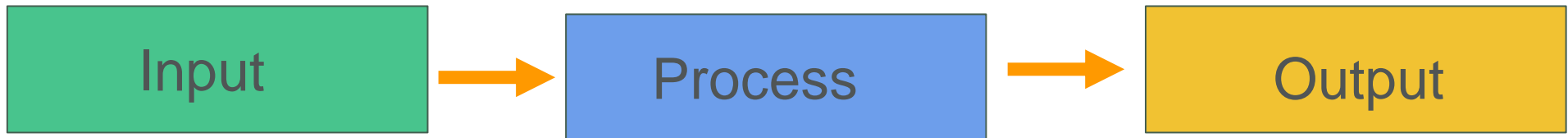


```
1 input.onGesture(Gesture.Shake, function () {
2   basic.showNumber(Math.randomRange(1, 6))
3 })
4
```



Computer – Systems

- + Input - Hardware that sends data to a computer system.
- + Output - Hardware used to communicate data that has been processed



Computer – Systems

The image displays the micro:bit programming environment. At the top, a blue header contains the 'micro:bit' logo, 'Home', 'Share', 'Blocks', and 'JavaScript' options. Below the header, a search bar is followed by a category list: Basic, Input, Music, Led, Radio, Loops, Logic, and Variables. The main workspace shows a script starting with an 'on shake' event block, followed by a 'show number' block containing a 'pick random' block set to '1 to 6'. An orange arrow points from the 'SHAKE' sensor on the micro:bit board to the 'on shake' block, labeled 'Input'. Another orange arrow points from the 'show number' block to the physical LEDs on the board, labeled 'Output'. A third orange arrow points from the 'pick random' block to the micro:bit board, labeled 'Processor'. The board image at the bottom shows internal components like the processor, compass, accelerometer, pins, USB antenna, reset, and battery.



Review

+ Computational thinking – Examples

- + Buy gift for your friend
- + Get the milk from grocery store
 - + Decomposition → Go to store, walk?, bike?, car?
 - + Abstraction → many aisles, which aisle to go to? Just milk, not candy
 - + Pattern recognition → Find milk aisle, cold section, dairy section
 - + Algorithms → Get cart, find milk, load milk, pay, load it to car, come home, put it in refrigerator
 - + Evaluation → Is it good or expired? One is enough? Can we make it better?

+ Programming

- + Selection
- + Iteration
- + Programming languages

+ Micro:bit – Introduction



References

+ Micro:bit Educational Foundation microbit.org

