كل ما يحتاجه الطالب في جميع الصفوف من أوراق عمل واختبارات ومذكرات، يجده هنا في الروابط التالية لأفضل مواقع تعليمي إماراتي 100 %

تطبيق المناهج الإماراتية | الاجتماعيات | الرياضيات

الصفحة الرسمية على التلغرام | الاسلامية | العلوم

الصفحة الرسمية على الفيسبوك | الانجليزية

التربية الاخلاقية لجميع الصفوف | اللغة العربية

التربية الرياضية

مجموعات التلغرام.	مجموعات الفيسبوك	قنوات تلغرام	قنوات الفيسبوك
الصف الأول	الصف الأول	الصف الأول	الصف الأول
الصف الثاني	الصف الثاني	الصف الثاني	الصف الثاني
الصف الثالث	الصف الثالث	الصف الثالث	الصف الثالث
الصف الرابع	الصف الرابع	الصف الرابع	الصف الرابع
الصف الخامس	الصف الخامس	الصف الخامس	الصف الخامس
الصف السادس	الصف السادس	الصف السادس	الصف السادس
الصف السابع	الصف السابع	الصف السابع	الصف السابع
الصف الثامن	الصف الثامن	الصف الثامن	الصف الثامن
الصف التاسع عام	الصف التاسع عام	الصف التاسع عام	الصف التاسع عام
الصف التاسع متقدم	الصف التاسع متقدم	الصف التاسع متقدم	تاسع متقدم
الصف العاشر عام	الصف العاشر عام	الصف العاشر عام	عاشر عام
الصف العاشر متقدم	الصف العاشر متقدم	الصف العاشر متقدم	عاشر متقدم
الحادي عشر عام	الحادي عشر عام	الحادي عشر عام	حادي عشر عام
الحادي عشر متقدم	الحادي عشر متقدم	الحادي عشر متقدم	حادي عشر متقدم
ثاني عشر عام	الثاني عشر عام	الثاني عشر عام	الثاني عشر عام
ثاني عشر متقدم	الثاني عشر متقدم	ثاني عشر متقدم	ثاني عشر متقدم

UNITED ARAB EMIRATES
MINISTRY OF EDUCATION

YEAR OF TOLERANCE

Design and Technology

Grade 6

Volume

1 2 3

Design and Technology

MAKER AND PYTHON

STUDENT BOOK

GRADE 6

VOLUME 3

Authored and designed by a specialised committee from the Ministry of Education

First edition 1439 - 1440 A.H. / 2018 - 2019 A.D.

"Extensive knowledge and modern science must be acquired. The educational process we see today is in an ongoing and escalating challenge which requires hard work.
We succeeded in entering the third millennium, while we are more confident in ourselves."

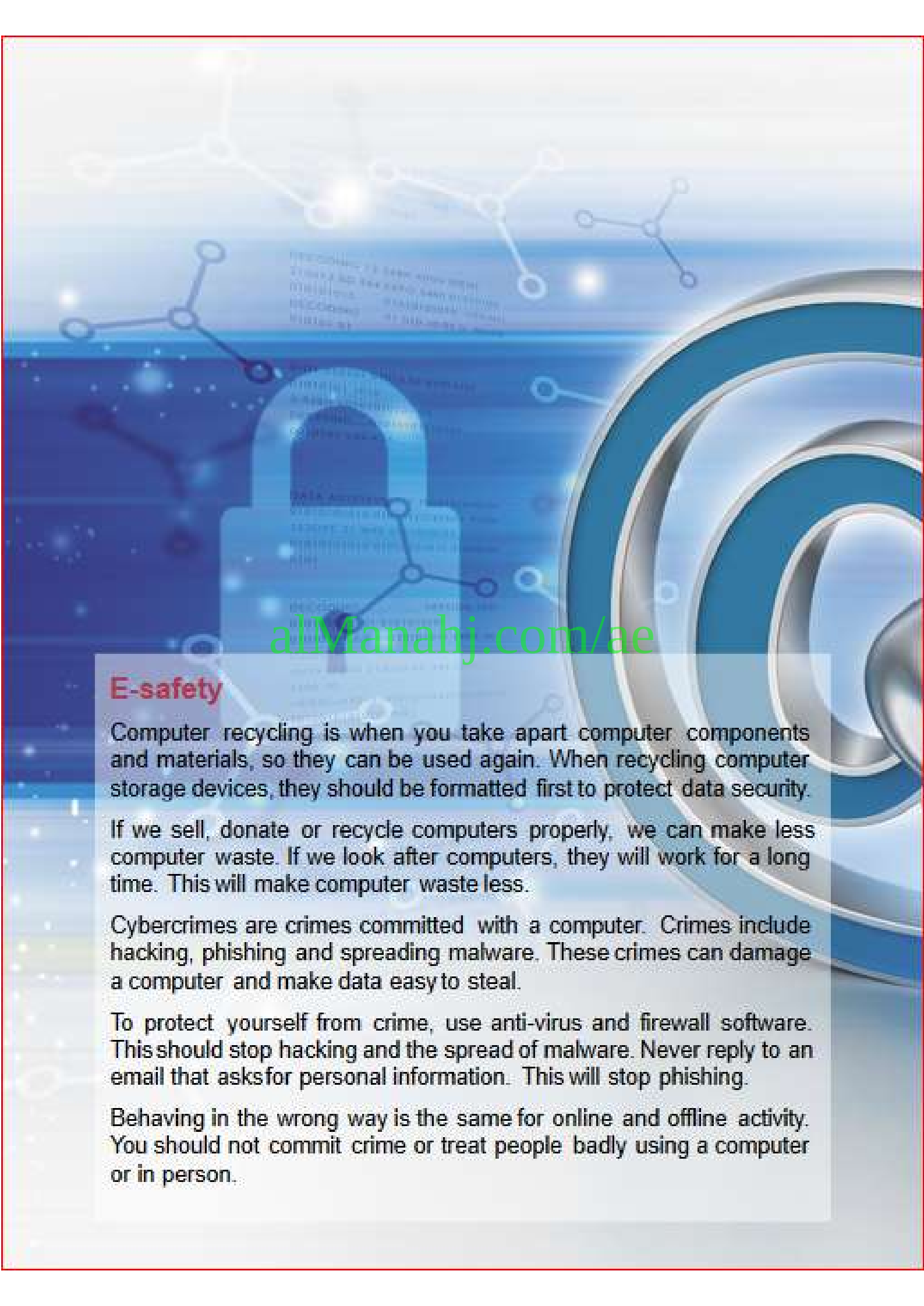
H.H. Sheikh Khalifa Bin Zayed Al Nahyan
President of the United Arab Emirates

Ministry of Education - Call Centre
For Suggestions, Inquiries & Complaints

80051115 | 04-2178855 | ccc.moe@moe.gov.ae | www.moe.gov.ae

Table of Contents

Table of Contents



E-safety

Computer recycling is when you take apart computer components and materials, so they can be used again. When recycling computer storage devices, they should be formatted first to protect data security.

If we sell, donate or recycle computers properly, we can make less computer waste. If we look after computers, they will work for a long time. This will make computer waste less.

Cybercrimes are crimes committed with a computer. Crimes include hacking, phishing and spreading malware. These crimes can damage a computer and make data easy to steal.

To protect yourself from crime, use anti-virus and firewall software. This should stop hacking and the spread of malware. Never reply to an email that asks for personal information. This will stop phishing.

Behaving in the wrong way is the same for online and offline activity. You should not commit crime or treat people badly using a computer or in person.

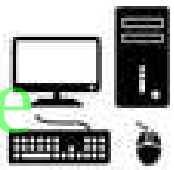
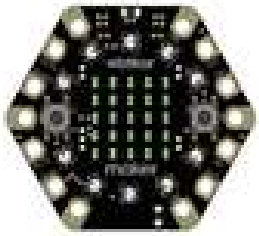
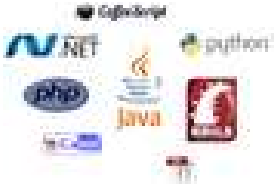
UNIT 1

Introduction to Makerand Python

Overview

In this unit, students will learn about graphical and text-based programming languages. They will discuss programs created in different languages. Students will look at the Maker and its hardware features again. Students will learn about hierarchy and abstraction. They will develop Python programming skills to download a simple program to the Maker using a USB cable.

Keywords

Term	Definition	Image
computer	a machine with input, processing and output devices	
Maker	a tiny programmable computer (microcontroller)	
programming	commands that solve a problem	
programming language	a language to create software programs	

Ardublockly	a graphical language to program Maker	
Python	a text-based programming language	

Learning outcomes

1.1 Understand the Maker (microcontroller) and Python software.

1.2 Understand that programming can be done using graphical and text-based languages.

1.3 Discuss different algorithms that solve the same problem. (G6.2.1.2.3)

1.4 Learn how to download programs to Maker using Python.

1.5 Describe the notion of hierarchy and abstraction in computing. (G6.2.3.1.2)

1.6 Identify an operating system and how it works. (G6.1.1.11.1)

Computer processing

The Maker is a microcontroller. It is a small computer. Before we learn about Maker, we are going to learn something about computer processing. This will help us understand how inputs, processing and outputs make computers work.

There are four functions that make computer processing happen:

1. **Inputs**– This is how a computer takes in information from the world. People get input through their senses (eyes and ears). A computer uses a mouse, keyboard or touchscreen as input.
2. **Processing** The CPU (Central Processing Unit) is a small microchip inside the computer. It's how the computer processes input and software instructions. You can think of the processor as the brains of a computer; the faster the processor, the more quickly the computer can think.
3. **Memory**– This is how the computer remembers things. There are two types of memory:

- RAM **(Random Access Memory)**- You can think of this as the computer's short-term memory while working.
- Storage **(harddrive)**- This is the computer's long-term memory. This is where a computer can store information even when the power is turned off.

4. **Outputs**– This is how a computer shows information after processing. People share information by talking. A computer uses text and graphics on a display.

Activity 1

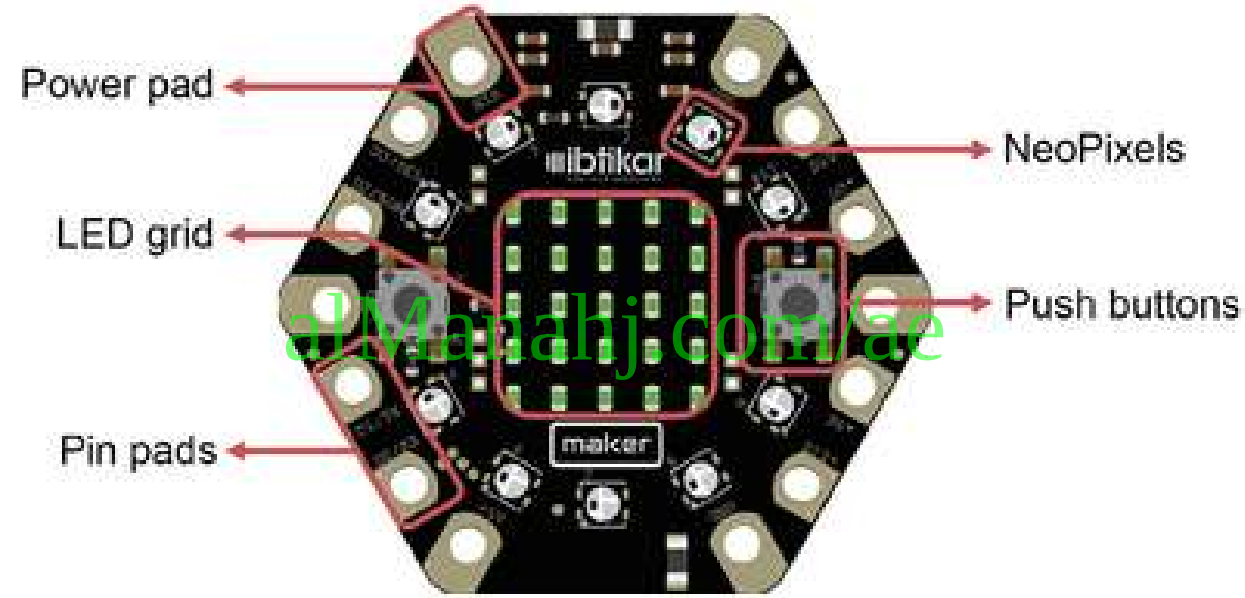
Match the computer function to its description.

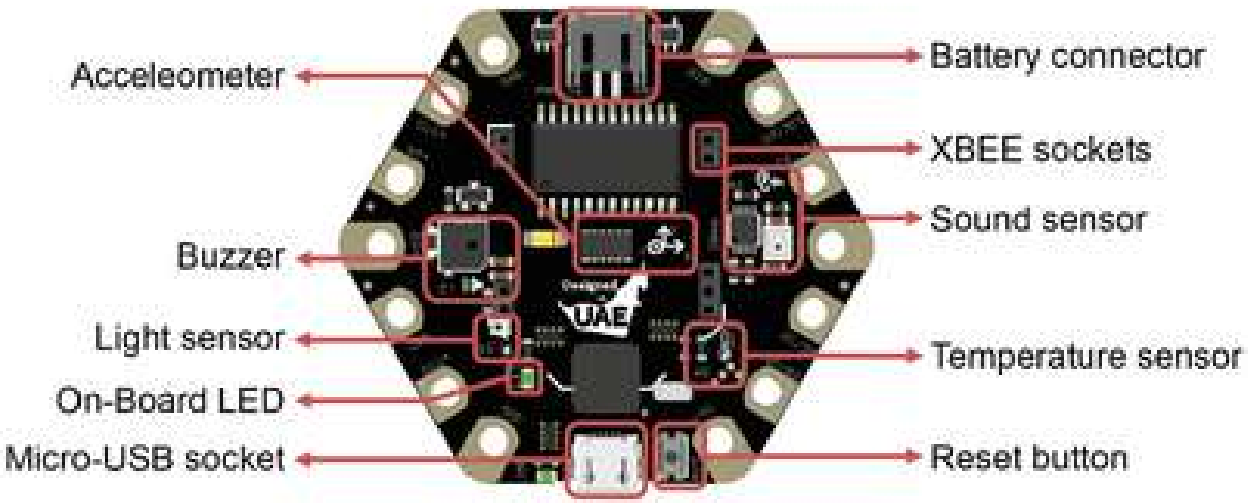
Function		Description
Input		This is how the computer processes input and software instructions.
Processing		This is how a computer takes in information from the world.
Memory		This is how a computer shows information after processing.
Output		This is how the computer stores information.

Introduction to Maker

We already know about the Maker microcontroller. A microcontroller is a small circuit board. It has a processor, memory and programmable hardware for inputs and outputs. These are built onto a single microchip.

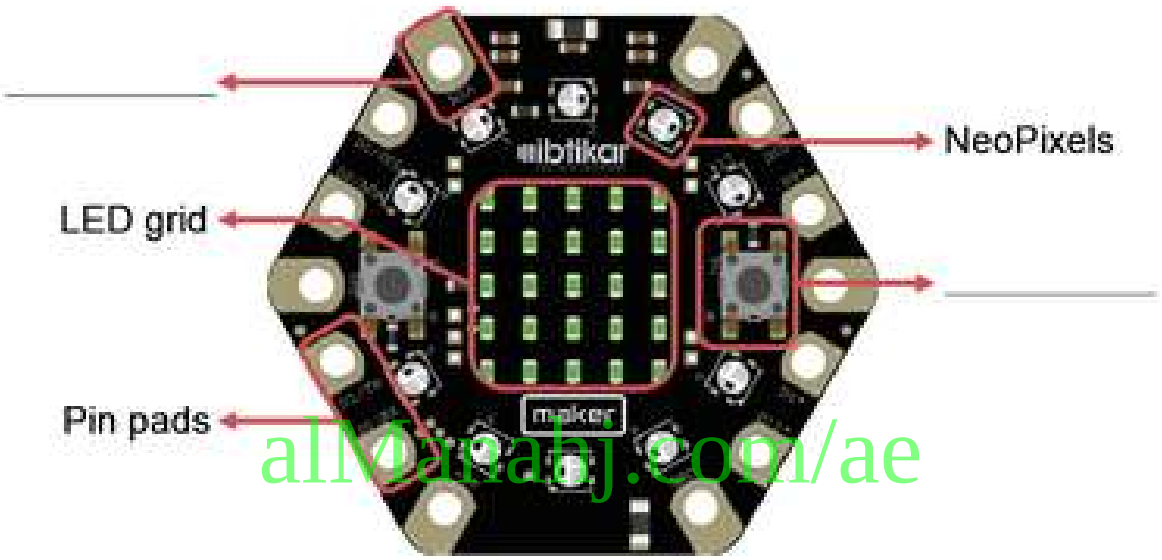
The Maker has push buttons, LED lights, RGB lights, an accelerometer and a speaker. Let's look at the hardware features on the Maker microcontroller again.

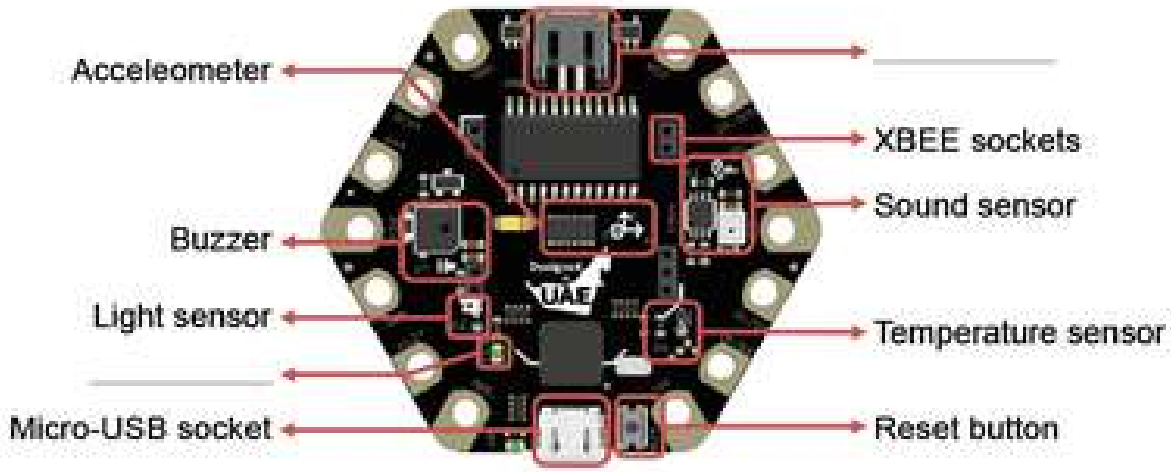




Activity 2

Look at the images. What are the Maker hardware features? Fill in the blank spaces.





Introduction to Python

The Maker is hardware. To use hardware, we need to create software (code or computer programs). The software tells the hardware what to do and when. This is so it can function.

We will be using Python to create software for Maker. Python is a simple text-based programming language. It is used by many organisations, such as YouTube and NASA.

The advantage of Python programming is that the words (syntax) of the language is simpler than other languages, such as C++. Python is still a powerful programming language. It can be used for many purposes.

The Python IDLE (integrated development and learning environment) is software used to write programs using Python code. The interface for the Python IDLE software looks like this:

Below is an example of a program written in Python. This program will output 'A' on the LED grid.

Python Example.py

File Edit Format Run Options Window Help

```
## ************************************************************
## Activity 5 - Show Characters on the LED Grid
##
## Arduino Sketch "SerialMaker.ino" Must Be Uploaded First
## ************************************************************

import time
from IBSerial import *
import IBSerial as IBMaker

IBMaker.Open_Port("COM13","115200") ## open the COM port
IBMaker.PinMAP("MAKER")             ## map pinMode
IBMaker.begin("V3.00")              ## or "V1.00"
## ************************************************************

while(1):
  IBMaker.Leds_Char('A', 1000)

## ************************************************************
IBMaker.Close_Port()                ## close the COM port
## ************************************************************
```

Ln: 1 Col: 0

Why Maker and Python?

Unlike most microcontrollers, the Maker was designed to be used as a tool to teach programming skills. This means students can create code (programs) with Python. Then they can see the programs working on the Maker hardware.

Students can get programming experience. They can do this by using simple text-based programming with physical inputs and outputs on the Maker. We cannot see how the program works if we are only using text-based programming.

Students can program the Maker for projects outside the computer lab. This is because the device can be carried everywhere.

Activity 3

Look at the sentences below. Fill in the blank spaces using the words in the table.

Python	physical	Maker	projects	text	program

Unlike most microcontrollers, the _____ was designed to be used as a tool to teach programming skills. This means students can create code (programs) with _____ Then they can see the programs working on the Maker hardware.

Students can get programming experience. They can do this using

simple text-based programming with _____ inputs and outputs on the Maker. We cannot see how the _____ works if we are only using _____-based programming.

Students can program the Maker for _____ outside the computer lab. This is because the device can be carried everywhere.

Graphical and text-based programming languages

We can use different programming languages and techniques to solve a computing problem. Programs can use different commands/syntax. Some programming algorithms work better than others.

- Programming languages like Ardublockly are **graphical** They use blocks for commands.
- Programming languages like Python are **text-based** They use written commands.

Discuss different algorithms that solve the same problem

Activity 4

Here are two examples of an algorithm. It outputs a happy face on the Maker.

Programming language	Program
Ardublockly	

Python	`import time`
	`from IBSerial import *`
	`import IBSerial as IBMaker`
	
	`IBMaker.Open_Port("COM13","115200") ## open the COM port`
	`IBMaker.PinMAP("MAJOR") ## map pinMode`
	`IBMaker.begin("V0.00") ## or "V1.00"`
	`## ...`
	
	`while(1):`
	`  IBMaker.Leds_Char('A', 1000)`

Discuss the algorithm examples above. Then, answer the following questions:

How are the algorithms similar? Write your answers in the box below.

How are the algorithms different? Write your answers in the box below.

Why is one algorithm better than the other?

How to download programs to the Maker using Python and Arduino software

Python IDLE needs to use **serial communication** to download programs to the Maker. This is different from Arduino and Ardublockly.

A teacher or technician will add two **library files** and a **sketch file** to the computer. This is so we can use Python to program the Maker.

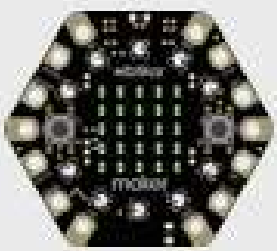
- **PySerial library** – This lets us set up Python for serial communication.
- **SerialMakeArduinoSketch** – This is downloaded to the Maker to set it up for serial communication.
- **IBSerial library** – This allows Arduino functions to be used with Python to program the Maker.

Programming Maker with Python

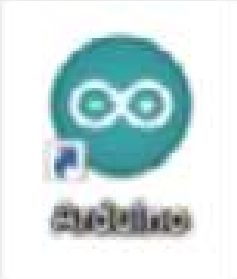
We will now practise programming the Maker by creating a program to output 'A'.

Activity 5

- Connect the Maker to the computer or laptop with the USB cable.



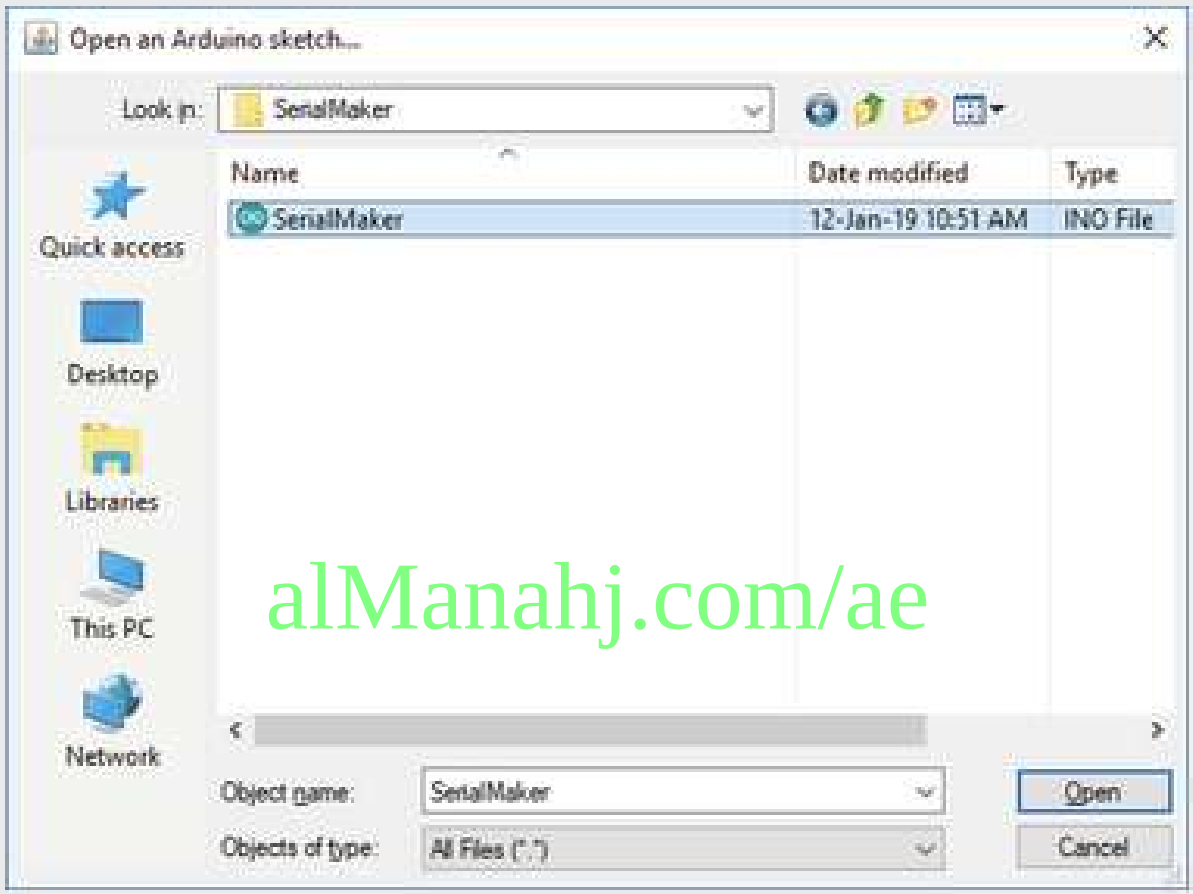
- Open the Arduino software by left double-clicking on a short-cut using the mouse. Or left-click on the program from the Start menu.



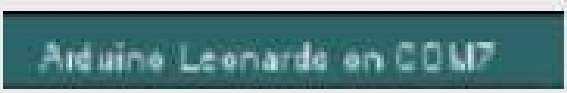
- The Arduino software should open showing the IDE interface.

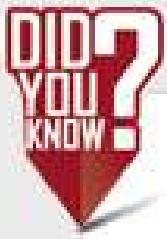
- Left-click on the open icon  by using the mouse. Or click File. Then, click Open from the menu.

- Using the mouse, left-click on the SerialMaker sketch. Then, click Open.



- Left-click on the upload icon by using the mouse. Or click Sketch. Then, click Upload from the menu.
- Once the sketch has uploaded successfully, the Maker can now be programmed using Python. Before closing the Arduino software, check the com port in the bottom right corner of Arduino IDE.





You only need to upload the SerialMaker sketch with Arduino once. This will configure the Maker. Then, you can write and download programs using Python.

- Open the Python IDLE software by left double-clicking on a shortcut using the mouse. Or left-click on the program from the Start menu.

- Type the following code into Python using the keyboard.

*Output A.py

File Edit Format Run Options Window Help

```
import time
from IBSerial import *
import IBSerial as IBMaker

IBMaker.Open_Port("COM13","115200") ## open the COM port
IBMaker.PinMAP("MAKER")             ## map pinMode
IBMaker.begin("V0.00")              ## or "V1.00"

while(1):
  IBMaker.Leds_Char('A', 1000)


IBMaker.Close_Port()                ## close the COM port
```

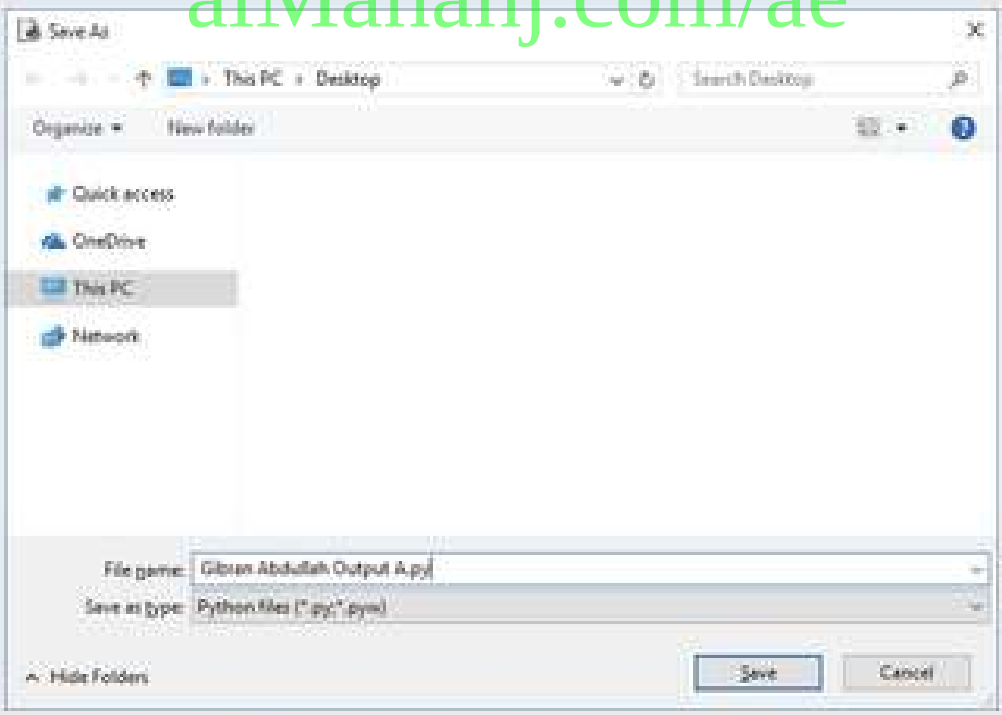
Ln: 16 Col: 0

- Check that line 4 uses the correct com port for your computer. This should match the com port as shown in Arduino:

E.g.

A com port (communication port) is a connection between a computer and another device. They connect serial devices. They send one bit of data at a time. For example, a mouse might be connected to COM1. The keyboard might be connected to COM2.

- Using the mouse, left-click File. Then, choose Save As. Give the file a sensible name, e.g. 'Gibran Abdullah Output A'. Then, click Save.



- To download the program to the Maker, press the F5 key on the keyboard. Or using the mouse, left-click Run. From the menu, click the Run module.

Challenge

add more code to the program to output 'B' and 'C'.

The Maker can only store one program at a time. When you download a new program to the Maker, it will replace anything you saved before.

Hierarchy and abstraction in computing

Hierarchy

In computing, we use a **hierarchy** to put software and hardware into levels. At the top level, users access software applications. Look at the other levels below:

Hardware/software level
• Application programs (e.g. Python IDLE)
• High-level programming language (e.g. Python)
• Low-level programming language
• Machine language
• Hardware/architecture

Using a hierarchy lets us work in one level without knowing about other levels.

- For example, we can use a **programming language** (e.g. Python) without knowing about a **machine language** or **architecture**.

Abstraction

In programming, **abstractions** are functions, classes or modules. A function is a set of commands to do a task. We collect functions into libraries, so they can be used again.

- For example, we can use the **LED character function** to output characters on the Maker LED grid.

```
IBMaker.Leds_Char('T', 1000)
```

- It is better to use the **characterfunction**This means you don't have to program each LED light to output a character.

```
IBMaker.setLed (0, 1, HIGH)
IBMaker.setLed (0, 2, HIGH)
IBMaker.setLed (0, 3, HIGH)
IBMaker.setLed (0, 4, HIGH)
IBMaker.setLed (0, 5, HIGH)
IBMaker.setLed (1, 3, HIGH)
IBMaker.setLed (2, 3, HIGH)
IBMaker.setLed (3, 3, HIGH)
IBMaker.setLed (4, 3, HIGH)
IBMaker.setLed (5, 3, HIGH)
```

Identify an operating system and how it works

An operating system(e.g. Microsoft Windows)is the software that manages computer hardware, software and resources. The operating system gives services for other software (e.g. Python IDLE)to work.

Activity 6

Look at the list of software. Is it an operating system? Write 'yes' or 'no'.

Softwarename	Operatingsystem?(Yesor No)
Microsoft Windows	
Python IDLE	
MAC OSX	
Microsoft Word	
Ardublockly	
Linux	

Unit 1 summary

In this unit, we:

- learned about graphical and text-based programming languages.
- looked at the Ibtikar Maker and its features again.
- were introduced to Python programming and IDLE software.
- learned about hierarchy and abstraction in programming.
- began to develop programming skills to create and download a program to Maker using Arduino, Python and a USB cable.

Pop quiz

Complete the pop quiz released by ADU.

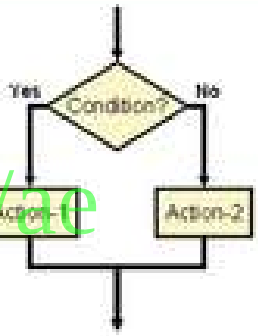
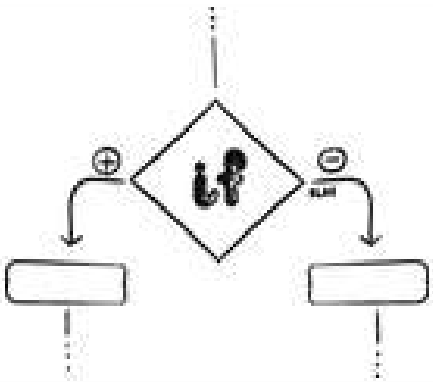
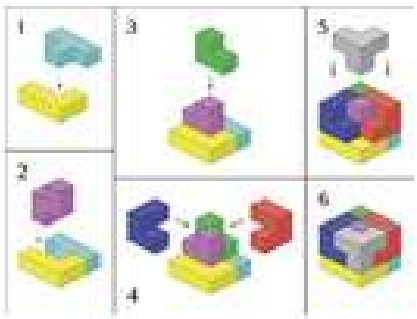
UNIT 2

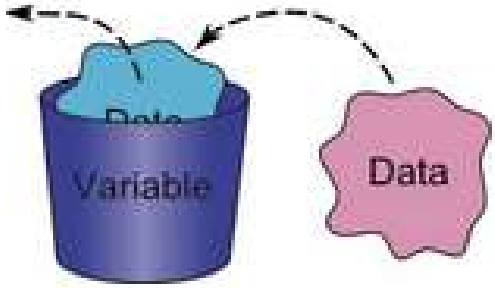
Selection with Python

Overview

In this unit, students will learn more about Python programming. This unit introduces logic with event handlers and conditional statements. These are used so a program can decide which tasks to do. Students will also learn how to use variables in Python. They will create programs that use them with logic.

Keywords

Term	Definition	Image
event handler	when an action is started by an event	
conditional statement	a decision-making process in programming	
algorithm	steps to solve a problem before programming	

comments	written notes to explain codes	
variable	a place to hold information	

Learning outcomes

2.1 Understand event handlers and when to use them in a program.

2.2 Understand conditional statements and when to use them in a program.

2.3 Understand what variables are and how to use them in a program.

2.4 Apply your knowledge by creating a Maker program that takes inputs and produces outputs using logic and variables.

Event handlers

In programming, an event is an action that is started by the user (e.g. a user pressing a key or clicking a mouse button).

- An event handler is the code that responds to the event.
- A programmer can write codes to tell the computer what to do when an event happens.

Below is an event handler in Ardublockly. In this example, pressing the button (Left A) is the event. The action is outputting 'A' on the LED grid.

Below is an event handler in Python. In this example, pressing the button (Left A) is the event. The action is outputting 'A' on the LED grid.

```
while(1):
  if (IBMaker.ButtonL()):
    IBMaker.Leds_Char('A', 200)
```

Activity 1

We will create a program that uses event handlers. The program will use the buttons as events. The action will be outputs on the LED grid.

Start by connecting the Maker to the computer. Upload the SerialMaker sketch in Arduino if you need to. Then, program the Maker in Python.

Event_Handlers.py

File Edit Format Run Options Window Help

```
## Arduino Sketch "SerialMaker.ino" Must Be Uploaded First

import time
from IBSerial import *
import IBSerial as IBMaker

IBMaker.Open_Port("COM13","115200") ## open the COM port
IBMaker.PinMAP("MAKER")             ## map pinMode
IBMaker.begin("V0.00")              ## or "V1.00"


while(1):
  if (IBMaker.ButtonL()):

  elif (IBMaker.ButtonR()):
    IBMaker.Leds_Num(6, 200)


IBMaker.Close_Port()                ## close the COM port
```

Ln: 1 Col: 0

Make sure you use the same com port as shown in Arduino. Then, save the file with a sensible name.

To download the program to the Maker, press the F5 key on the keyboard. Or using the mouse, left-click Run. From the menu, click the Run module.

Challenge

add the code to output 'A' on the LED grid when button (Left A) is pressed.

Variables

Computer programs use information. To do this, a program will store information using variables and constants.

- Programmers create variables to store information that can change.
- Programmers create constants to store information that will not change.

Variables can store many types of information. We will use two types of variables in Python:

- Text
- Number

Below is an example of a number variable in Python:

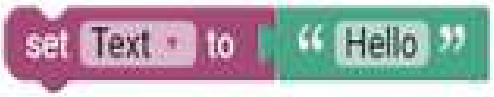
```
Height = 1.70
```

Below is an example of a text variable in Python:

```
Country = "UAE"
```

Activity 2

Match the variables to their data type.

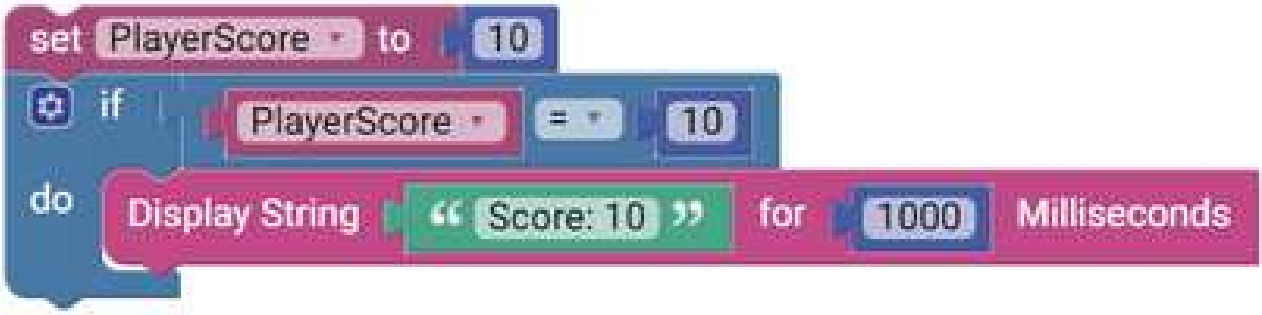
Variable		Datatype
`Name = "Mohammed"`		Text
		
`Age = 5`		Number
		

Conditional statements

Computer programs are instructions. They tell a computer how to process input and give output. Part of programming is telling the computer 'WHEN' to do an action.

- A conditional statement is logic. It decides when to do an action. These are similar to event handlers.
- Sometimes a conditional statement is called an If-Then statement. IF a condition is met, THEN an action is done.

Below is a conditional statement in Ardublockly. In the example, the condition is score equals 10. The action is outputting 'Score: 10' on the LED grid.



Below is a conditional statement in Python. In the example, the condition is score equals 10. The action is outputting 'Score: 10' on the LED grid.

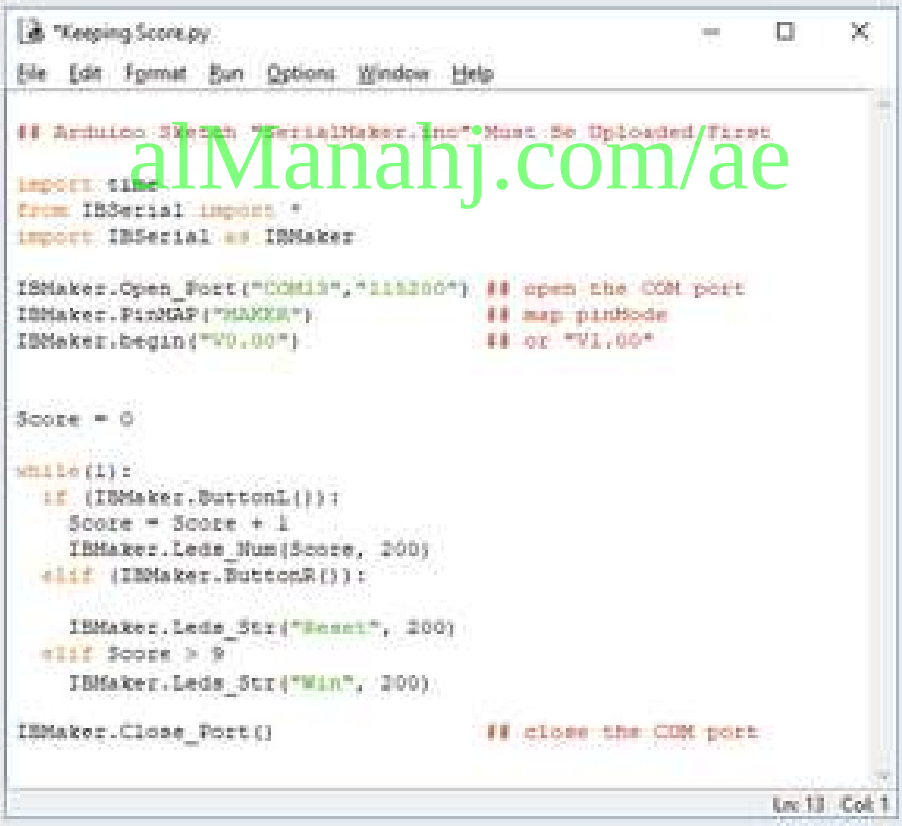
```
  while(1):
score = 10
if (score == 10):
  IBMaker.Leds_Str("Score: 10", 1000)
```

Activity 3

We will create a program that uses event handlers, variables and conditional statements.

- The program will use the buttons as events.
- The actions will be adding to or resetting the score variable.
- The conditional will output 'Win' when the score is more than 9.

Start by connecting the Maker to the computer. Upload the SerialMaker sketch in Arduino if you need to. Then, program the 'Keeping Score' program in Python.



Make sure you use the same com port as shown in Arduino. Then, save the file with a sensible name.

To download the program to the Maker, press the F5 key on the keyboard. Or using the mouse, left-click Run. From the menu, click the Run module.

Challenge

add the code below to set the score to 0 when button (Right B) is pressed.

```
Score = 0
```


Activity 4

Match the code to the descriptions.

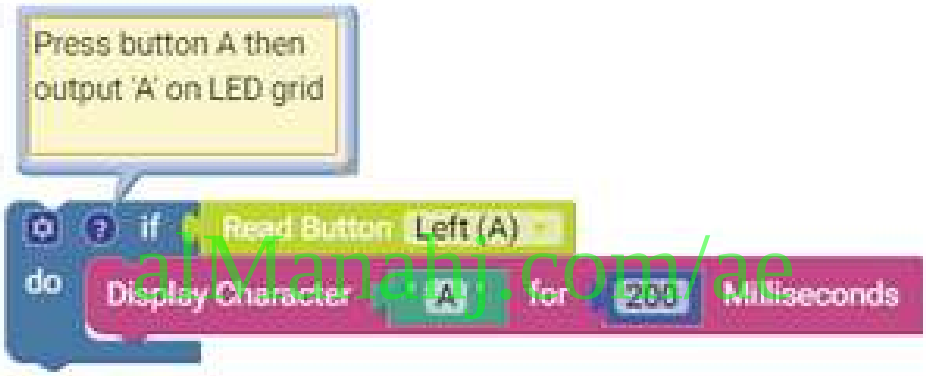
Code		Description
`elif Score > 0`		**Variable**
`if (IBMaker.ButtonL()):`		**Eventhandler**
`Score = 0`		**Conditional statement**

Commenting on your code

When programming, it is good practice to add comments to your code:

- Comments help you remember what the code does.
- Comments help others to understand the program.

Below is an example of a comment in Ardublockly:



To add a comment in Python, we use a # symbol. Below is an example of a comment in Python:

```
IBMaker.Leds_Chr('A', 200)    #Press button A Then output 'A' on LED grid
```

Activity 5

Practise adding comments to the code. One comment has been done for you.

Comments to use:

Pressbutton Right B. Set the score to 10 and output 'Reset' on the LED grid.
Pressbutton Right B. Set the score to 0 and output 'Reset' on the LED grid.
Pressbutton Left A. Add 1 to the score and output it on the LED grid.
When the score is more than 9, output 'Win' on the LED grid.
Set the score to 0.

Code	Comment
`Score = 0`	

`if (IBMaker.ButtonL()):`
`  Score = Score + 1`
`  IBMaker.Leds_Num(Score, 200)`
	Press button Left A. Add 1 to the score and output it on the LED grid.

`elif (IBMaker.ButtonR()):`
`  Score = 0`
`  IBMaker.Leds_Str("Reset", 200)`	

`elif Score > 9` `  IBMaker.Leds_Str("Win", 200)`	

Activity 6

Now we know how to write comments. We should add comments to the code block in Python.

Open the 'Keeping Score' program in Python. Then, add the comments from Activity 5 using a # symbol.

```
  Score = 0                    #Comment...

while(1):

  if (IBMaker.ButtonL()):      #Comment...
    Score = Score + 1
    IBMaker.Leds_Num(Score, 200)

  elif (IBMaker.ButtonR()):    #Press button Left A. Add 1 to score thenoutput on LED grid
    Score = 0
    IBMaker.Leds_Str("Reset", 200)

  elif Score > 9               #Comment...
    IBMaker.Leds_Str("Win", 200)

IBMaker.Close_Port()           ## close the COM port
```


REMEMBER

Savethe file after you have added the comments.

Unit 2 summary

In this unit, we

- introduced logic with event handlers and conditional statements.
- learned how to use variables in Python.
- developed Python skills to create a program that uses variables, event handlers and conditional statements.

End of unit quiz

Date	

1	An event handler is code that responds to an event.	
	A	True
	B	False

2	How do we use variables?	
	A	Inputting information
	B	Storing information
	C	Processing information
	D	Outputting information

3	Sometimes a conditional statement is called _____.	
	A	If-Then
	B	If-When
	C	If-So
	D	If-What

4	Which code did you use to output Score on the Maker LED grid?	
	A	Score = 0
	B	(IBMaker.ButtonR()):
	C	IBMaker.Leds_Num (Score, 200)
	D	elif Score > 9

5	Which symbol do we need for comments in Python?	
	A	#
	B	*
	C	@
	D	$

Notes

Notes

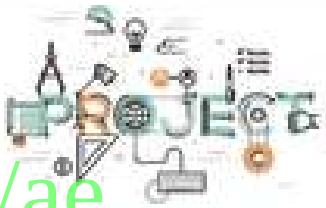
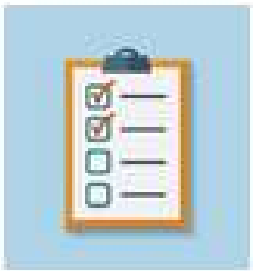
UNIT 3

Makerand PythonProject

Overview

Students will show their understanding and programming skills learned in Units 1 and 2 in the Term 3 project. The project program will use logic and variables to process different inputs and send them to different outputs. Students will also self-evaluate the project. They will think and talk about outcomes with their peers and the teacher.

Keywords

Term	Definition	Images
project	assessed work with many tasks	
task	one part of the assessed work in a project	
requirements	things you need to do or have, to do well in an activity or project, e.g. has features of a 3D model	
musical instrument	a device used to make music, e.g. piano	

Learning outcomes

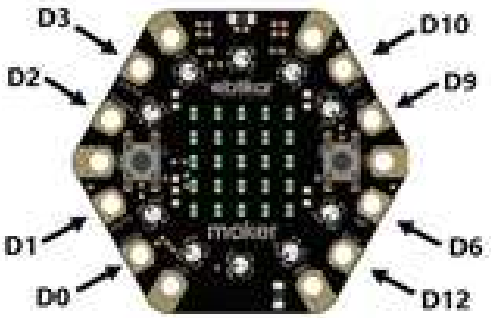
3.1 Identify the key elements of the problem, such as inputs, outputs, assumptions and limitations. (G6.2.1.1.1)

3.2 Demonstrate an understanding of variables, event handlers and conditional statements by using them in the project.

3.3 Test the project program and evaluate your performance to identify areas for improvement.

About pin pads

The Maker has a set of 8 pin pads around its edges. The pin pads can be used to connect to other hardware for input or output. The pin pads are labelled D0, D2, etc. They can be programmed individually.



We can program the pin pads to work as a button using an event handler. To do this, we use read capacitive.

- Read capacitive knows when there is a change in electrical charge (capacitance) of a pin pad.
- The electrical charge (capacitance) changes when a person touches the pin pad.
- Touchscreens on mobile phones also work by knowing when there are changes in capacitance.

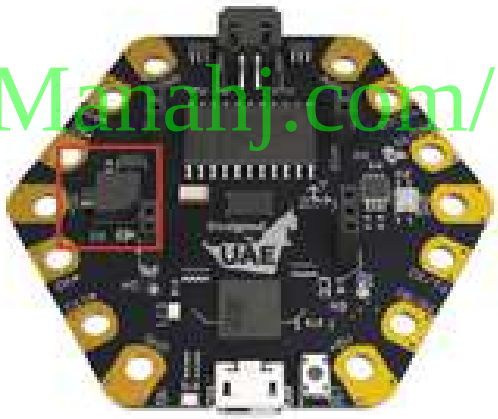
Below is a read capacitive event handler in Ardublockly. In the example, the event is touching the pin pad (D0). The action is outputting an 'A' on the LED grid.

Below is a read capacitive event handler in Python. In the example, the event is touching the pin pad (D0). The action is outputting an 'A' on the LED grid.

```
if (IBMaker.Touch(0) > 200):
    IBMaker.Leds_Char('A', 200)
```

About the buzzer

The buzzer is a component that can output sounds from the Maker. We can change the sound using different frequencies. We can also change the duration (how long something happens).



Below is an example of sound output in Ardublockly. In the example, 100 is the frequency. Duration is 250 milliseconds.

Below is an example of sound output in Python. In the example, 100 is the frequency. Duration is 250 milliseconds.

Project

Project Brief

For this project, you will use the Maker like a piano.

First, you will plan how to program the Maker. You will match inputs with the outputs. Use the requirements to help you. After this, you will create your program with Python. Then, you will test your program. After testing, you will evaluate your work.

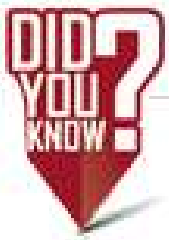
The piano program will use:

- variables to store the capacitive and tone values.
- the push button and conditional statement to change tone frequency.
- the LED grid to output the tone setting (A or B).
- pin pads as event handlers for each sound.
- the buzzer to output each sound.

Program sounds on the buzzer. Use the frequencies below.

Pinpad	Frequency
D3	200
D2	350
D1	500
D0	650
D12	800
D6	950
D9	1100
D10	1250

We don't have a musical keyboard. We can use crocodile clips and coins with the Maker pin pads. You can use each coin as a piano key.



If you don't have coins, you can use anything conductive (lets electricity go through it). Examples are tin foil, paper clips or even a banana!

Look at the requirements for the piano program.

Basicrequirements

1. Set T and C variables at the start of the program.
2. Input button (Left A) sets T variable to (1).
3. Input button (right B) sets T variable to (1.15).
4. Input pin pad (D3) outputs (T * 200) on the buzzer.
5. Input pin pad (D2) outputs (T * 350) on the buzzer.

Advancedrequirements

6. Input pin pad (D1) outputs (T * 500) on the buzzer.
7. Input pin pad (D0) outputs (T * 650) on the buzzer.
8. Input pin pad (D12) outputs (T * 800) on the buzzer.
9. Input pin pad (D6) outputs (T * 950) on the buzzer.
10. Input pin pad (D9) outputs (T * 1100) on the buzzer.

Activity 1

Now you have looked at the **projectbrief** and the piano program **requirements**. Talk about what you need to do for the project.

Use the box to write your ideas.

Understanding the project brief

Answer the questions below. This will show that you understand the Maker piano project.

Activity 2

1.	For this project, you will use the Maker like a piano.
a	True
b	False

2.	What will we use for input in the project program?
a	Buzzer
b	NeoPixels
c	LED grid
d	Pin pads

3.	What will we use for output in the project program?
a	Buzzer
b	NeoPixels
c	LED grid
d	Pin pads

4.	What will we use the code (below) for in the project program? `if (IBMaker.ButtonR()):`
a	Input
b	Thoroughput
c	Output
d	Not used

5.	What will we use the code (below) for in the project program? `IBMaker.playTone(I * 200, 100)`
a	Input
b	Processing
c	Output
d	Not used

Planning

Plan your program. First, read the descriptions. Then, match the descriptions and inputs with the outputs. Three examples have been done for you.

Activity 3

Description	Input	Output
Input button (Left A) sets T variable to (1). This outputs A on the LED grid.	`if (IBMaker.ButtonR()):`	`T = 1` `IBMaker.Leds_Char('A', 200)`
Input button (Right B) sets T variable to (1.15). This outputs B on the LED grid.	`if (IBMaker.ButtonR()):`	`IBMaker.playTone(T * 200, 100)`
Input pin pad (D3) outputs (T * 200) on the buzzer.	`if (IBMaker.Touch(3) > C):`	`T = 1.15` `IBMaker.Leds_Char('B', 200)`
Input pin pad (D2) outputs (T * 350) on the buzzer.	`if (IBMaker.Touch(2) > C):`	`IBMaker.playTone(T * 650, 100)`

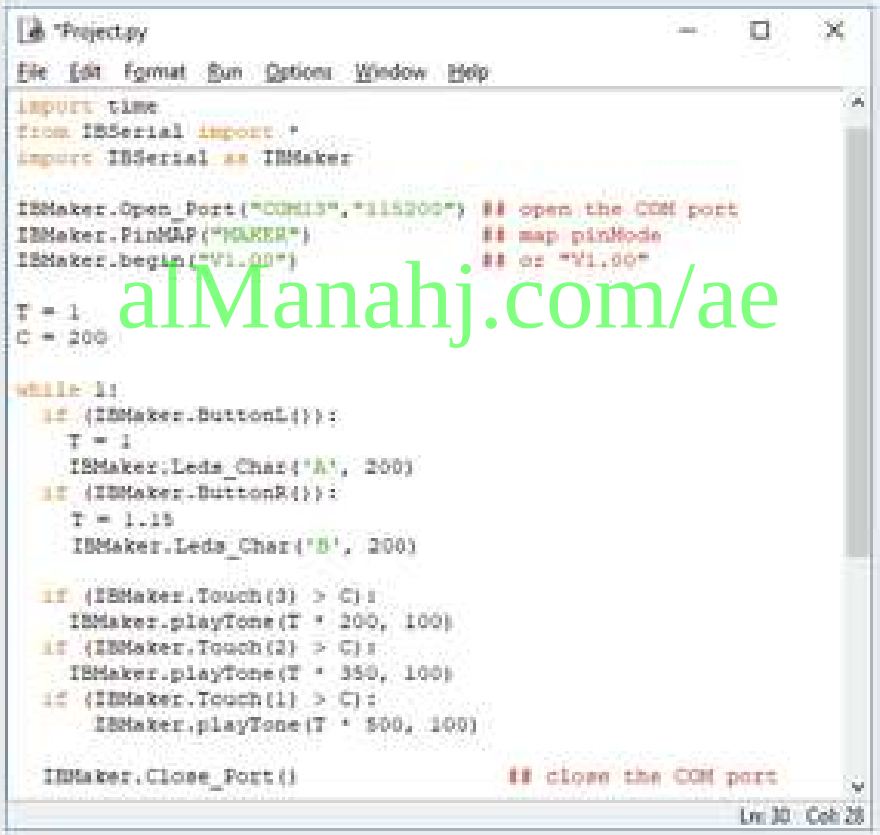
Input pin pad (D1) outputs (T * 500) on the buzzer.	`if (IBMaker.Touch(1) > C):`		`IBMaker.playTone(T * 500, 100)`
Input pin pad (D0) outputs (T * 650) on the buzzer.	`if (IBMaker.Touch(0) > C):`		`IBMaker.playTone(T * 350, 100)`
Input pin pad (D12) outputs (T * 800) on the buzzer.	`if (IBMaker.Touch(12) > C):`		`IBMaker.playTone(T * 800, 100)`
Input pin pad (D6) outputs (T * 950) on the buzzer.	`if (IBMaker.Touch(6) > C):`		`IBMaker.playTone(T * 950, 100)`

Programming

Create the project program. Use the program plan.

Activity 4

Connect the Maker to the computer. Upload the SerialMaker sketch in Arduino if you need to. Then, program the Maker in Python.



Make sure you use the same com port as shown in Arduino. Then, save the file with a sensible name.

To download the program to the Maker, press the F5 key on the keyboard. Or using the mouse, left-click Run. From the menu, click the Run module.

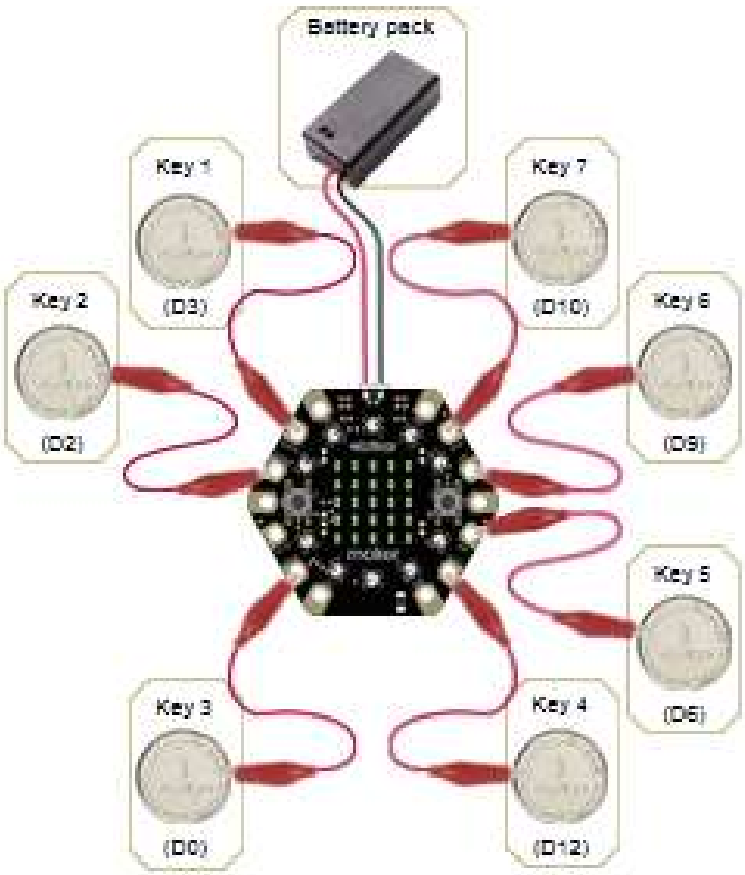
Challenge

Add more code to the program. The other pin pads should play the buzzer with these frequencies:

Pinpad	Frequency
D0	650
D12	800
D6	950
D9	1100
D10	1250

Makerpianoschematic

Use the Maker piano schematic to connect crocodile clips and coins. Then, test the program by touching each coin.



Studentworkbox

It is important for you to save your work. It can be used for evidence and feedback in the future.

You should take a screenshot of the program you have built for your Maker piano. You should save it on your computer. Print it out and stick it in the box below.

Testing

After the programming stage, you have to test it.

Test your Maker piano program. Does it meet the requirements of the project? If the Maker does not do what you programmed it to do, go back and try to fix it.

Activity 5

Test	Result	
Are the T and C variables set at the start of the program?	☐ No	☐ Yes
Does input button (Left A) set T variable to (1)?	☐ No	☐ Yes
Does input button (Left B) set T variable to (1.15)?	☐ No	☐ Yes
Does input pin pad (D3) output (T * 200) on the buzzer?	☐ No	☐ Yes
Does input pin pad (D2) output (T * 350) on the buzzer?	☐ No	☐ Yes

Did you answer No to any questions? If so, can you change your program to make your test score better?

Did you make changes to your program? Why?

Self-reflection

When you have finished your project, you have to think about how you worked. Answer the questions below. They will help you know where you did well and where you can do better.

Activity 6

Read the sentences below. Tick the box that is for you [√].

Sentences	I need more help.	I got better during the project.	I am good!
I could understand the project brief and its requirements.	☐	☐	☐
I correctly matched inputs to their outputs.	☐	☐	☐
I was able to use Python to create my program.	☐	☐	☐
I was able to build a Maker piano using the schematic.	☐	☐	☐
I did my own testing and fixed any mistakes.	☐	☐	☐

Are there any other areas you could make better?

Teacher evaluation rubric

Section		Criteria				
Brief (5) 1 mark for each of the criteria met		One mark for each question answered correctly (as per answers provided in teacher guidance)				
Planning (5) 1 mark for each of the criteria met		One mark for each of the inputs that were correctly matched to their respective outputs as per the basic and advanced requirements provided.				
Programming 1 mark for each of the following requirements met	**Basic** requirements (5)	T and C variables were set at the start of the program.	Input button (Left A) set T variable to (1).	Input button (Left A) set T variable to (1.15).	Input pin pad (D3) output (T * 200) on the buzzer.	Input pin pad (D2) output (T * 350) on the buzzer.
	Advanced requirements (5)	Input pin pad (D1) output (T * 500) on the buzzer.	Input pin pad (D0) output (T * 650) on the buzzer.	Input pin pad (D12) output (T * 800) on the buzzer.	Input pin pad (D6) output (T * 950) on the buzzer.	Input pin pad (D9) output (T * 1100) on the buzzer.
Testing (5) 1 mark for each of the criteria met		One mark for each test conducted verifying that the basic requirements had been met.				
Self-reflection (5) 1 mark for each of the criteria met		Honestly evaluated ability to understand the brief and requirements	Honestly evaluated ability to correctly match inputs with outputs	Honestly evaluated ability to use Python to create a program	Honestly evaluated ability to build the Maker piano	Honestly evaluated ability to conduct the verification process

Project brief	+	Planning	+	Programming	+	Testing	+	Self-reflection	=	Total mark
/5		/5		/10		/5		/5		/30

Teacher feedback box

Your teacher will give you feedback on your work.

What went well	Even better if

Comments: feedback and future targets for students

Unit 3 summary

In this unit we:

- showed our understanding of programming and the skills learned in Units 1 and 2.
- connected crocodile clips to make pin pads easier to use for input.
- learned to use the buzzer for output.
- created a project program (using logic and variables) to process different inputs and send them to different outputs.
- tested the program.
- thought about our skills and how we worked.

UNIT

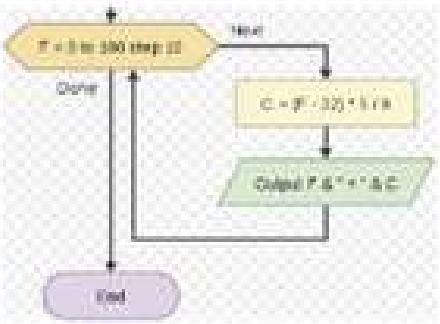
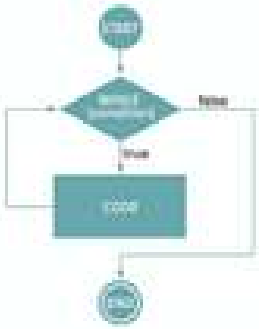
4

Iterationwith Python

Overview

Students will look at the theory of iteration again. They will learn how to use loops in Python. Students will work on changing long programs (with code that repeats) to short programs. Students will also understand some common functions that they can create using loops.

Keywords

Term	Definition	Images
looping	to repeat program commands over and over again	
for loop	a type of loop used in programming	
while loop	to repeat instructions until a condition is met	

Learning objectives

4.1 Understand the concept of iteration.

4.2 Practise using looping to make programs more efficient.

4.3 Perform common editing and formatting functions in word processing. (G6.1.2.8.1)

4.4 Insert, edit and format tables in a document. (G6.1.2.8.2)

Introduction to looping

In computer programming, we use looping to repeat a sequence of code. Looping is also called **repetition**

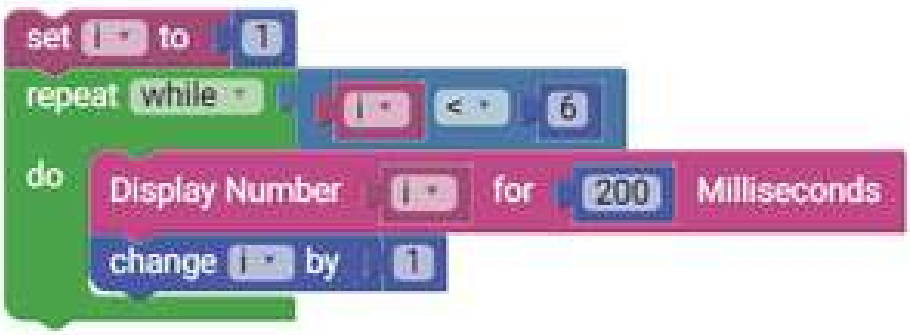
- A counted loop uses a variable to repeat a number of instructions.
- A conditional loop is used to repeat until a condition is met.

Below is a counted loop in Ardublockly. In this example, the loop will output a count from 1 to 5:

Below is a counted loop in Python. In this example, the loop will output a count from 1 to 5:

```
for i in range(1,5):
  IBMaker.Leds_Num(i, 100)
```

Below is a conditional loop in Ardublockly. In this example, the loop will output a count from 1 to 5:



Below is a conditional loop in Python. In this example, the loop will output a count from 1 to 5:

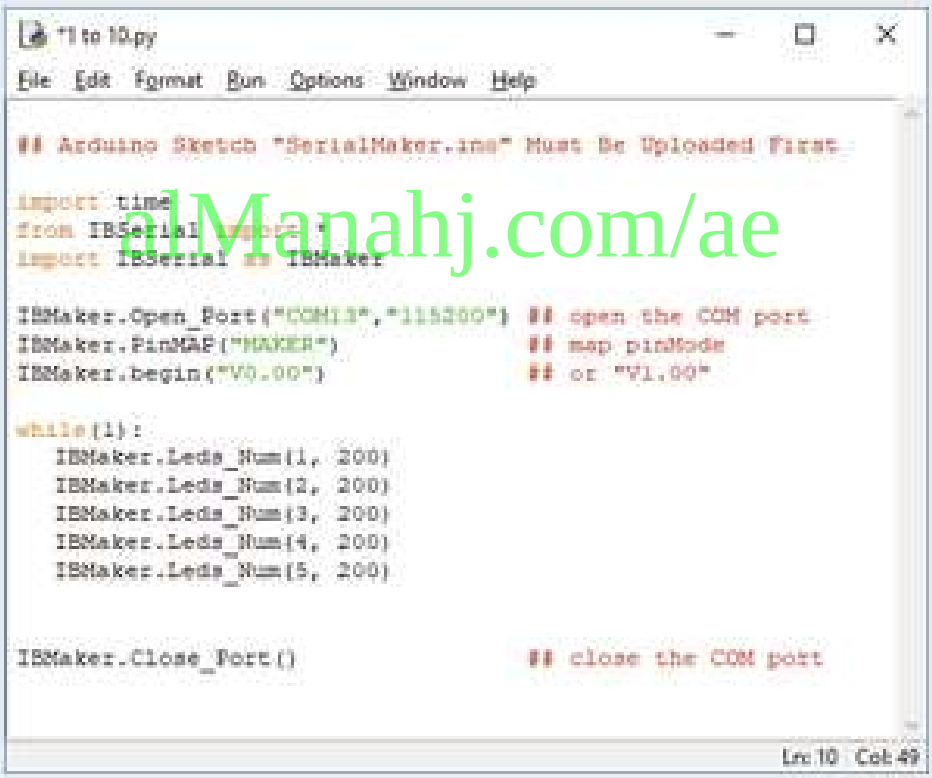
```
i = 1
while (count < 6):
 IBMaker.Leds_Num(i, 200)
 i = i + 1
```

Usingloopingto makeprogramsworkbetter

Activity 1

Before we use loops, we will create a program that outputs the numbers from 1 to 5 in sequence.

Connect the Maker to the computer. Upload the SerialMaker sketch in Arduino if you need to. Then, program the Maker in Python:



Make sure you use the same com port as shown in Arduino. Then, save the file with a sensible name.

To download the program to the Maker, press the F5 key on the keyboard. Or using the mouse, left-click Run. From the menu, click the Run module.

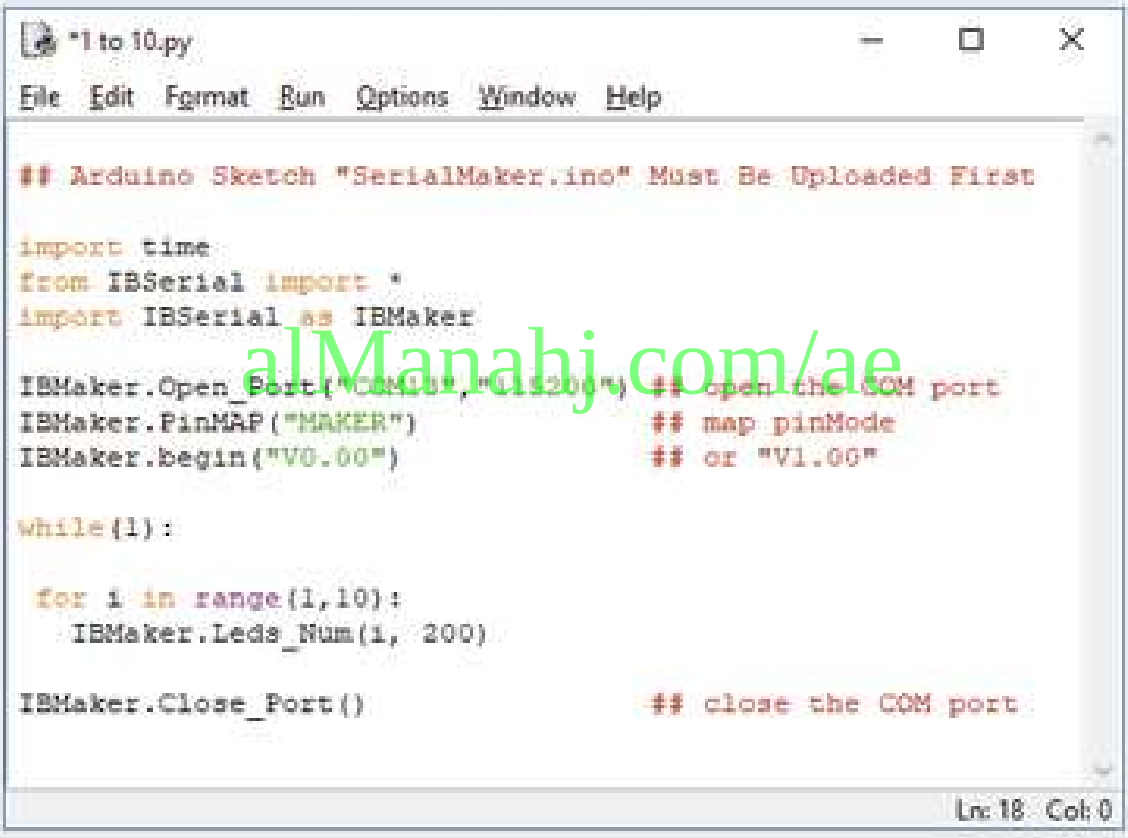
Challenge

Add code to output numbers 6 to 10.

Activity 2

Create a new program to count from 1 to 10. We will use a loop to repeat similar instructions. This program will work better.

Connect the Maker to the computer. Upload the SerialMaker sketch in Arduino if you need to. Then, program the Maker in Python:



Make sure you use the same com port as shown in Arduino. Then, save the file with a sensible name.

To download the program to the Maker, press the F5 key on the keyboard. Or using the mouse, left-click Run. From the menu, click the Run module.

Challenge Countto 20

Change the program to count to 20.

Challenge Countdown

Change the program to count down from 10 to 1.

Year of Tolerance

2019 is the Year of Tolerance in the UAE. We should create a program to celebrate!

Editing and formatting in word processing

First, we will plan the program to celebrate Tolerance using MS Word. The program will:

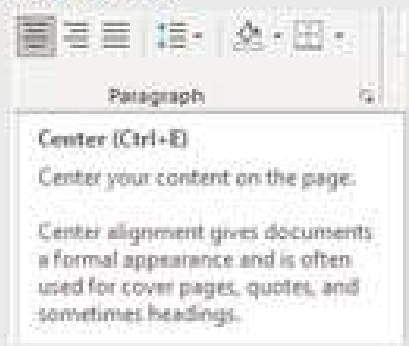
- output important words, e.g. UAE and Tolerance.
- use the UAE colours on the NeoPixels, e.g. red and green.

Activity 3

Open Microsoft Word. Add the title 'UAE Year of Tolerance'.

Select the text. Then, make it **bold**

Then, place your title in the centre of the page. Use the **Center** alignment.

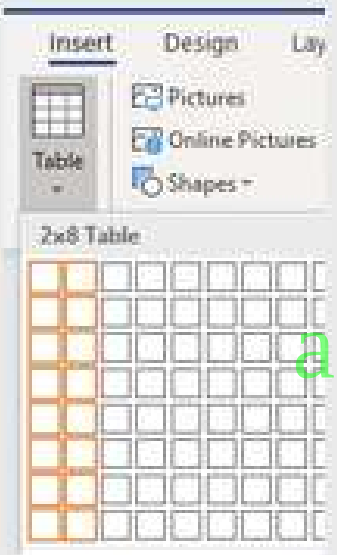


Now use left alignment and bullet points to make a list of words and colours about the Year of Tolerance. We could output them in our program.

Save the document with a sensible name.

Activity 4

Open the Word document where you started to plan the Year of Tolerance program. Then, insert a table with two columns and eight rows.



Add titles and numbers to your table.

Number	Instruction
1	
2	
3	
4	
5	
6	
7	

Which instructions do you want to use? Choose from the list below. Add them to your table. This will help us plan how the program will work.

Instructions

- Output 'UAE' on the LED grid.
- Output 'Tolerance' on the LED grid.
- Output 'Welcoming' on the LED grid.
- Output 'Happiness' on the LED grid.
- Output 'Safety' on the LED grid.
- Output 'Prosperity' on the LED grid.
- Light up the NeoPixels red.
- Light up the NeoPixels green.
- Light up the NeoPixels white.

Finally, add a design to the table from Table Tools.

Save the document with a sensible name.

Activity 5

Now we will use your plan to create the Year of Tolerance program.

Connect the Maker to the computer. Upload the SerialMaker sketch in Arduino if you need to. Then, start to program the Maker in Python:

*Tolerance.py

File Edit Format Run Options Window Help

```
## Arduino Sketch "SerialMaker.ino" Must Be Uploaded First

import time
from IBSerial import *
import IBSerial as IBMaker

IBMaker.Open_Port("COM13","115200") ## open the COM port
IBMaker.PinMAP("MAKER")             ## map pinMode
IBMaker.begin("V0.00")              ## or "V1.00"

while(1):



IBMaker.Close_Port()                ## close the COM port
```

Ln: 16 Col: 57

Now add the code for your instructions.

Instruction	Code
Output 'UAE' on the LED grid.	`IBMaker.Leds_Str("UAE", 200)`
Output 'Tolerance' on the LED grid.	`IBMaker.Leds_Str("Tolerance", 200)`
Output 'Welcoming' on the LED grid.	`IBMaker.Leds_Str("Welcoming", 200)`

Output 'Happiness' on the LED grid.	`IBMaker.Leds_Str("Happiness", 200)`
Output 'Safety' on the LED grid.	`IBMaker.Leds_Str("Safety", 200)`
Output 'Prosperity' on the LED grid.	`IBMaker.Leds_Str("Prosperity", 200)`

Light up the NeoPixels red.	`IBMaker.clearPixels()`
	`for i in range(0,9):`
	`  IBMaker.setPixelColor(i, IBMaker.colorWheel(255, 0, 0))`
	`  time.sleep(0.250)`

Light up the NeoPixels green.	`IBMaker.clearPixels()`
	`for i in range(0,9):`
	`  IBMaker.setPixelColor(i, IBMaker.colorWheel(0, 255, 0))`
	`  time.sleep(0.250)`

Light up the NeoPixels white.	`IBMaker.clearPixels()`
	`for i in range(0,9):`
	`  IBMaker.setPixelColor(i, IBMaker.colorWheel(255, 255, 255))`
	`  time.sleep(0.250)`

Make sure you use the same com port as shown in Arduino. Then, save the file with a sensible name.

To download the program to the Maker, press the F5 key on the keyboard. Or using the mouse, left-click Run. From the menu, click the Run module.

Unit 4 summary

In this unit, we:

- learned about the theory of iteration.
- learned how to use loops in Python.
- changed long programs (with code that repeats) to short, better programs with loops.
- learned about some common functions that we created with loops.

Pop quiz 2

Date

1
A
B
C

2
A
B
C

3
A
B
C

4
A
B
C

5
A
B
C

Evaluation	
Student evaluation	Teacher's evaluation
★ ★ ★ ★ ★	/ 5

UNIT 5

Makerand PythonMini Project

Overview

Students will show their understanding and programming skills learned in Units 1 to 4 in a mini project. The project will use logic, variables and iteration to process different inputs and send them to different outputs. Students will also evaluate and present their project. They will think and talk about outcomes with their peers and the teacher.

Keywords

Term	Definition	Images
mini project	work with small tasks	
task	one part of the assessed work in a project	
requirements	things you need to do or have, to do well in an activity or project, e.g. has features of a 3D model	

Learning outcomes

5.1 Demonstrate an understanding of variables, event handlers, conditional statements and iteration by using them in a mini project.

5.2 Test the mini project program and evaluate your performance to identify areas for improvement.

5.3 Interact with peers employing a variety of digital environments and media. (G6.1.4.3.3)

5.4 Develop complex multimedia presentation. (G6.1.2.7.1)

Fidget cube mini project

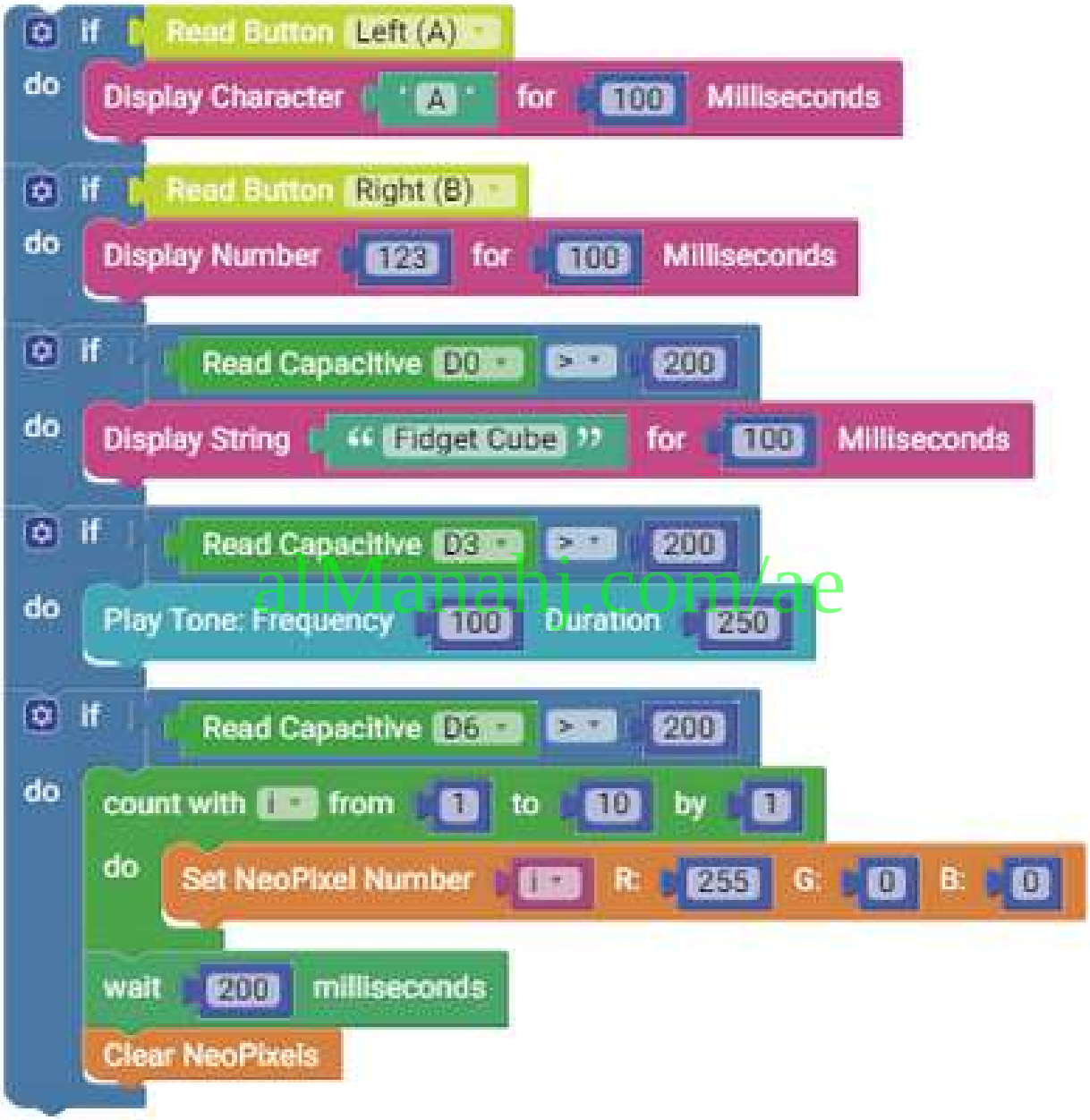
A fidget cube is a small object with buttons, switches and dials. People who like to 'fidget' (keep moving your hands, feet or body) find it relaxing to push, press and play.

For this mini project, you will program your Maker as a 'fidget cube.'

The program will:

- use 5 or more inputs (e.g. push buttons and pin pads).
- use 5 or more outputs (e.g. LEDs, NeoPixels and a buzzer).
- use 5 functions (inputs matched with outputs).

Below is an example of a fidget cube program in Ardublockly:



Here are inputs and outputs for the fidget cube program.

Inputs

A	`if (IBMaker.ButtonL()):`
B	`if (IBMaker.ButtonR()):`
C	`if (IBMaker.Touch(0) > item):`
D	`if (IBMaker.Touch(3) > item):`
E	`if (IBMaker.Touch(6) > item):`
F	`if (IBMaker.Touch(9) > item):`

Outputs

1	`IBMaker.Leds_Chr('A', 100)`
2	`IBMaker.Leds_Num(123, 100)`
3	`IBMaker.Leds_Str("Fidget Cube", 100)`
4	`IBMaker.playTone(100, 250)`

5	`for i in range(0,9):`
	`  IBMaker.setPixelColor(i, IBMaker.colorWheel(255, 0, 0))`
	`time.sleep(0.200)`
	`IBMaker.clearPixels()`

6	`for i in range(0,9):`
	`  IBMaker.setPixelColor(i, IBMaker.colorWheel(0, 255, 0))`
	`time.sleep(0.200)`
	`IBMaker.clearPixels()`

Planningthefidgetcubeprogram

Activity 1

Now we will plan the fidget cube program functions. Match inputs with outputs. Then, describe each function.

No.	Input	Output	Whatwillthisdo?
1.	**Example:** A	**Example:** 1	**Example:** When the push button (Left A) is pressed, it will output 'A' on the LED grid.
2.			
3.			
4.			
5.			
6.			

Activity 2

Now we will use your plan to create the fidget cube program.

Connect the Maker to the computer. Upload the SerialMaker sketch in Arduino if you need to. Then, start to program the Maker in Python:

*Fidget Cube.py

File Edit Format Run Options Window Help

```
## Arduino Sketch "SerialMaker.ino" Must Be Uploaded First

import time
from IBSerial import *
import IBSerial as IBMaker

IBMaker.Open_Port("COM13","115200") ## open the COM port
IBMaker.PinMAP("MAKER")             ## map pinMode
IBMaker.begin("V0.00")              ## or "V1.00"

item = 200

while 1:

  if (IBMaker.Button1()):
    IBMaker.Leds_Chr('A', 100)



IBMaker.Close_Port()                ## close the COM port
```

Ln: 22 Col: 0

Challenge

Use your plan. Add the code for your functions to the program.

Make sure you use the same com port as shown in Arduino. Then, save the file with a sensible name.

To download the program to the Maker, press the F5 key on the keyboard. Or using the mouse, left-click Run. From the menu, click the Run module.

Fidgetcubeprojecttesting

After programming, you should test your fidget cube program.

Activity 3

Test	Result	
Does it use push buttons for input?	☐ No	☐ Yes
Does it use pin pads for input?	☐ No	☐ Yes
Does it use the LED grid for output?	☐ No	☐ Yes
Does it use a buzzer for output?	☐ No	☐ Yes
Does it use the NeoPixels for output?	☐ No	☐ Yes