

1

# Teaching examples of applications of programming in real life

Mr LIU Man Cheong  
Head of Computer Subject and  
STEM Education Coordinator  
Lai King Catholic Secondary School



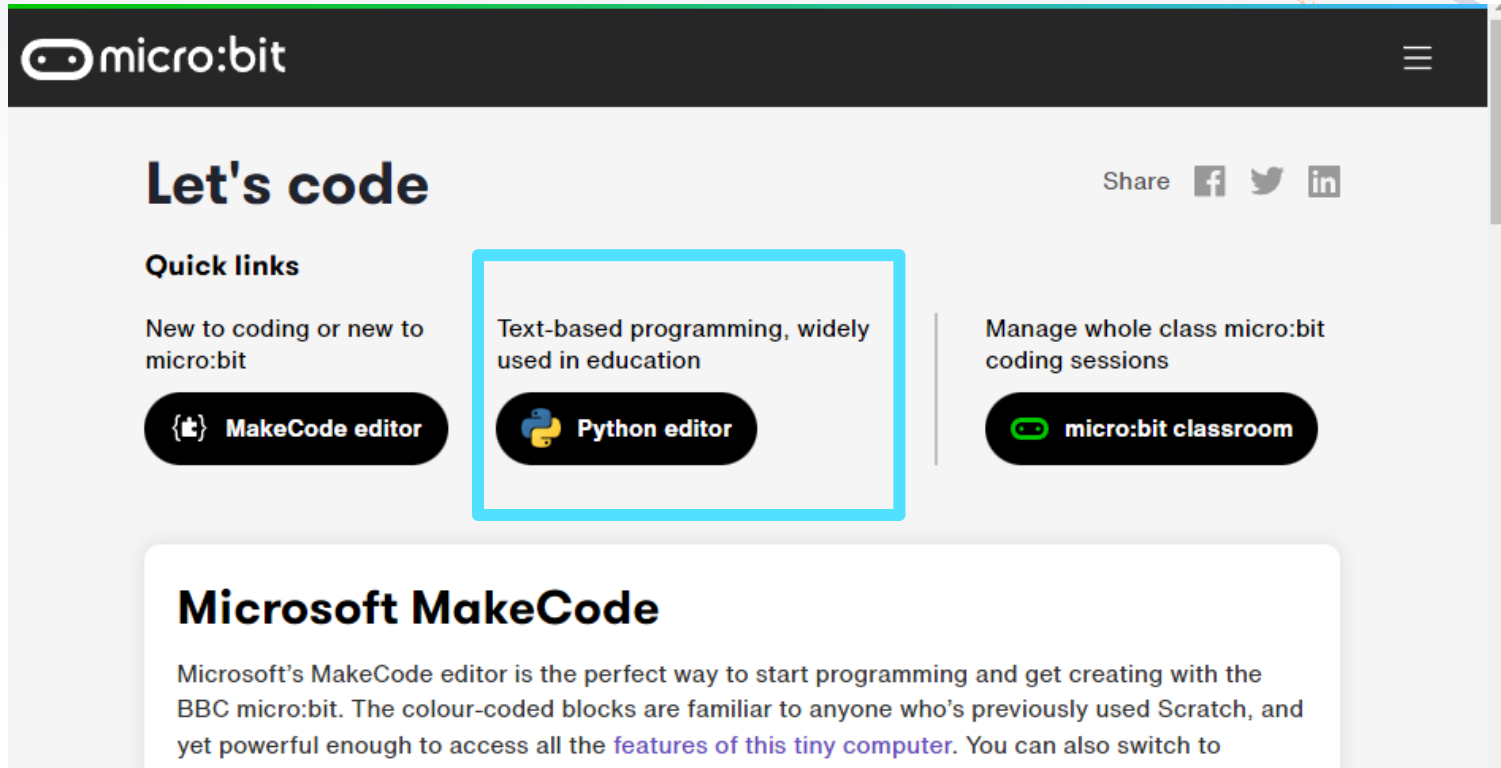


**b. Applications of Programming in Real Life (6 hours)**

	<ul style="list-style-type: none"><li>• Use extended programming modules or libraries in writing programs to interact with physical devices.</li></ul>	<p>Students should be able to use extended modules or libraries for capturing data from sensors (e.g. light sensor and accelerometer) and controlling specific devices (e.g. motor).</p> <p>Details of extended modules or libraries are not required.</p>
	<ul style="list-style-type: none"><li>• Use event handlers in writing event-driven programs.</li><li>• Construct simple programs on physical devices by using features/components of physical devices like speech recognition and accelerometer.</li></ul>	<p>Specific events include user actions (e.g. pressing a button) and sensor values (e.g. the reading from the light sensor is over a defined value).</p> <p>Details of event handlers are not required.</p> <p>Examples include generating a text display by speech recognition, controlling the movement of motors and detecting motion by accelerometer.</p>

# (A) Micro:bit




<https://microbit.org/code/>



The screenshot shows the Micro:bit website homepage. At the top left is the Micro:bit logo. Below it is the heading "Let's code" and a "Share" button with social media icons for Facebook, Twitter, and LinkedIn. Under "Quick links", there are three buttons: "MakeCode editor", "Python editor" (highlighted with a blue box), and "micro:bit classroom". The "Python editor" button is described as "Text-based programming, widely used in education". Below this is a section for "Microsoft MakeCode" with a description of the editor.


**micro:bit**


## Let's code

Share   

**Quick links**


New to coding or new to micro:bit

 **MakeCode editor**

 **Python editor**

Text-based programming, widely used in education

Manage whole class micro:bit coding sessions

 **micro:bit classroom**

## Microsoft MakeCode

Microsoft's MakeCode editor is the perfect way to start programming and get creating with the BBC micro:bit. The colour-coded blocks are familiar to anyone who's previously used Scratch, and yet powerful enough to access all the [features of this tiny computer](#). You can also switch to

<https://python.microbit.org/v/2>

The screenshot shows the Python Microbit v2 web editor. At the top, there is a navigation bar with icons for Download, Connect, Load/Save, Open Serial, and Help. A search bar labeled 'Script Name' contains the text 'microbit program'. Below the navigation bar is a code editor with the following Python code:

```
1 # Add your Python code here. E.g.
2 from microbit import *
3
4
5 while True:
6     display.scroll('Hello, World!')
7     display.show(Image.HEART)
8     sleep(2000)
9
```

<https://python.microbit.org/v/beta>

The screenshot shows the Python Microbit v/beta web editor. The interface is split into two main sections. On the left is a 'Reference' sidebar with a search bar and a list of categories: Variables, Display, Buttons, and Loops. On the right is the code editor for an 'Untitled project' with the following Python code:

```
1 # Imports go at the top
2 from microbit import *
3
4
5 # Code in a 'while True:' loop repeats forever
6 while True:
7     display.show(Image.HEART)
8     sleep(1000)
9     display.scroll('Hello')
10
```

- ▶ There are also some offline editors: E.g. Mu  
<https://codewith.mu/>

# Demo 1 - Button Event in Python

## Make a click-counter

The number on the display increases by 1 when Button A is pressed.



<https://scorecount.com/click-counter/>

# Demo 1 - Button Event in Python

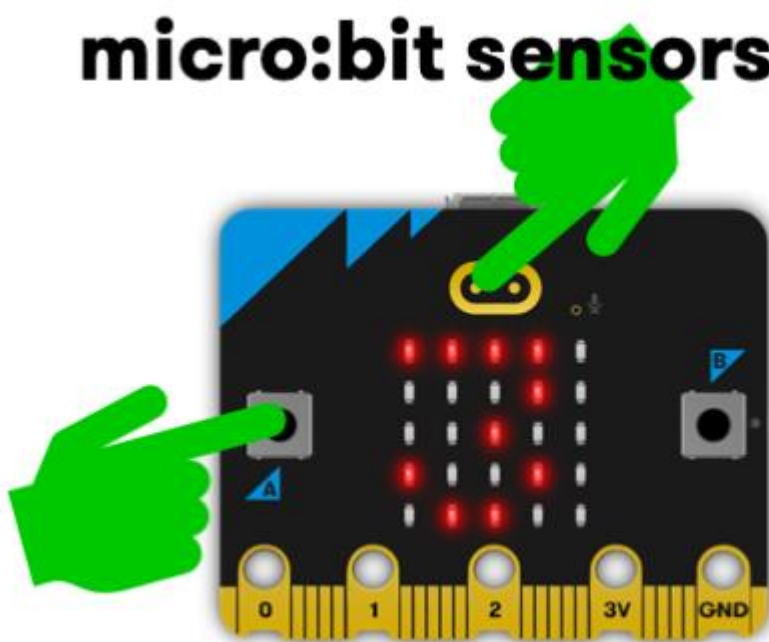
```
1 # Imports go at the top
2 from microbit import *
3
4 # On start
5 img = Image('00000:'
6             '0904:'
7             '09040:'
8             '90009:'
9             '09990')
10
11 display.show(img)
12 sleep(400)
13 count = 0
14
15 # Code in a 'while True:' loop repeats forever
16 while True:
17     if button_a.is_pressed():
18         count = count + 1
19         display.scroll(count)
20
21
```

**No “On Start” code block**  
because the code will  
execute sequentially.

A while:True infinite loop  
resembles the “Forever”  
code block

# Demo 2 - read data from sensors in Micro:bit v2

## micro:bit sensors

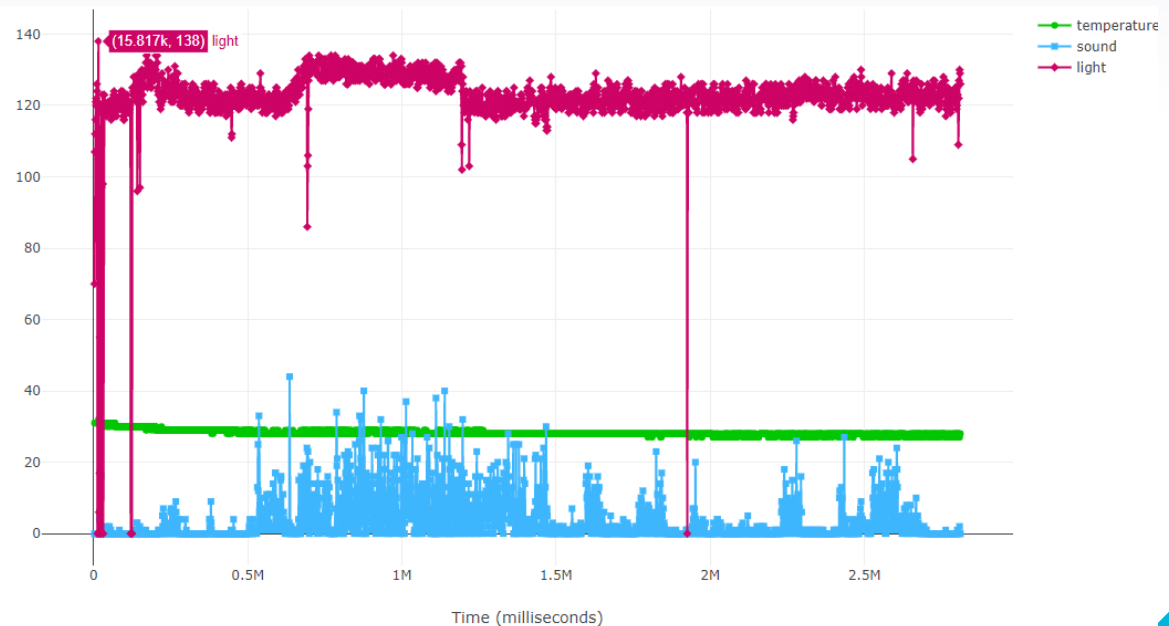


<https://microbit.org/get-started/first-steps/sensors/>

# Demo 2 - read data from sensors in Micro:bit v2

- Log the temperature, sound and light data
- Plot a graph

Time (milliseconds)	temperature	sound	light
3160	31	0	70
4168	31	0	107
5301	31	0	112
6392	31	0	116
7400	31	0	121
8409	31	0	120
9417	31	0	122
10609	31	0	126
11616	31	0	124
12624	31	0	0
13632	31	0	0
14724	31	0	121
15817	32	0	138
16825	31	0	105
17832	31	0	6
18840	32	4	0





# Demo 2 - read data from sensors in Micro:bit v2

```
1 # Imports go at the top
2 from microbit import *
3 import log
4
5 log.set_labels('temperature', 'sound', 'light')
6 display.show(Image.HAPPY)
7 sleep(400)
8
9 # Code in a 'while True:' loop repeats forever
10 while True:
11     l = display.read_light_level()
12     s = microphone.sound_level()
13     t = temperature()
14
15     print(l,s,t)
16
17     log.add({'temperature': t,
18            'sound': s,
19            'light': l
20            })
21     sleep(1000)
22
```

## Solutions

# (B) Arduino

**TINKERCAD**  
AUTODESK  
Tinkercad

課程 圖庫 部落格 學習 教授



Man Cheong Liu

搜尋設計...

3D 設計

電路

程式碼區塊

課程

您的課程

集合

+ 建立集合

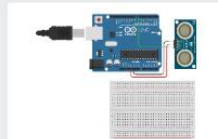


Educators: If you've ever heard "This is cool! What's next?"<sup>x</sup> while teaching with Tinkercad, here's how to guide your students to the next step when they're ready for professional - grade tools. [Learn more](#)

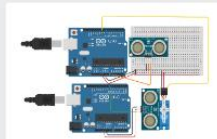
## Circuits

建立新電路

Select



Tremendous Robo  
2 小時前  
私人



Amazing Esboo-Snicket  
2 小時前  
私人



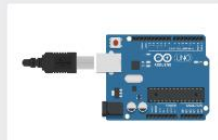
Brilliant Gaaris  
1 年前  
私人



Magnificent Lappi-Bruticus  
1 年前  
私人



Mighty Leelo-Juttuli  
1 年前  
私人



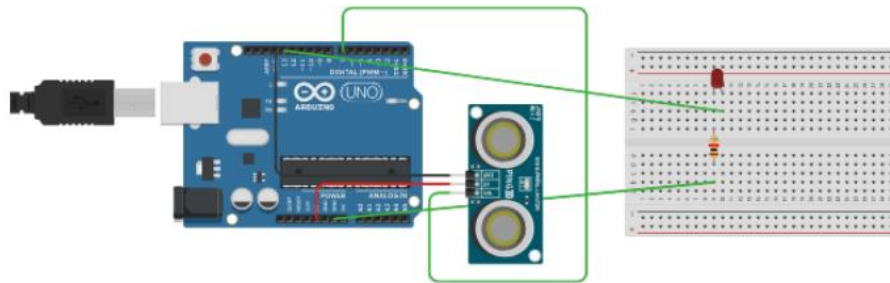
Powerful Fulffy  
2 年前  
私人

[Autodesk Tinkercad  
https://www.tinkercad.com/](https://www.tinkercad.com/)

- No hardware is required! 😊
- Students can work at home by their PCs
- But the UI does not support tablets very well. 😞

## (B) Arduino

<https://www.tinkercad.com/things/eU5pgW878rd-ultrasonic-sensor>



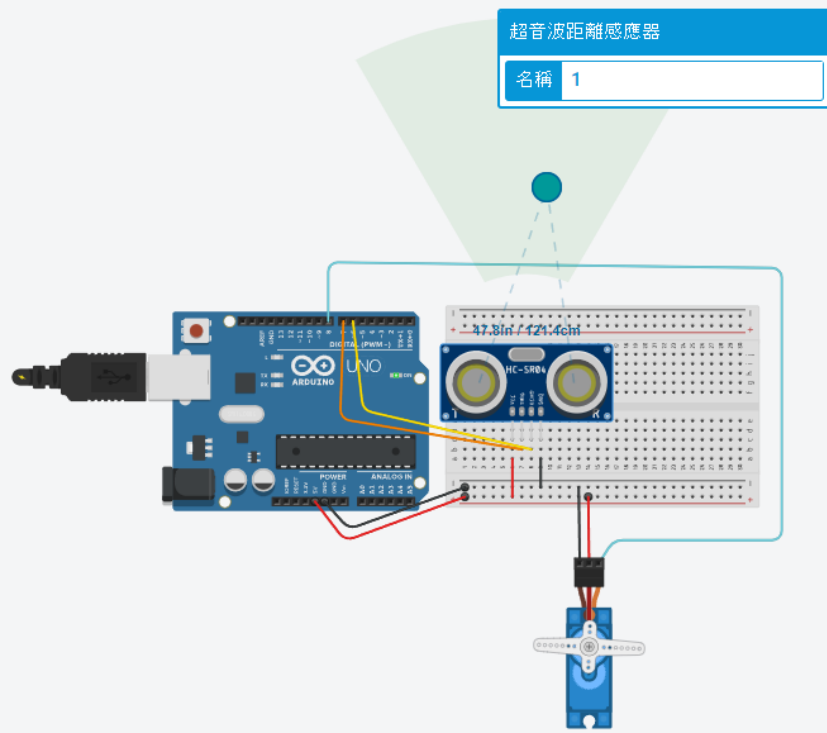
- Many code references.
- Feel free to tinker

## Demo 3 – Car park gate

- The car park gate will be automatically opened when the car distance is lower than 100cm



[https://commons.wikimedia.org/wiki/File:HK\\_TW\\_%E8%8D%83%E7%81%A3\\_Tsuen\\_Wan\\_%E8%95%99%E8%8D%83%E8%B7%AF\\_Wai\\_Tsuen\\_Road\\_near\\_%E7%B6%A0%E6%A5%8A%E6%96%B0%E9%82%A8\\_Luk\\_Yeung\\_Sun\\_Chuen\\_carpark\\_exit\\_gate\\_May\\_2020\\_SS2\\_09.jpg](https://commons.wikimedia.org/wiki/File:HK_TW_%E8%8D%83%E7%81%A3_Tsuen_Wan_%E8%95%99%E8%8D%83%E8%B7%AF_Wai_Tsuen_Road_near_%E7%B6%A0%E6%A5%8A%E6%96%B0%E9%82%A8_Luk_Yeung_Sun_Chuen_carpark_exit_gate_May_2020_SS2_09.jpg)



[https://www.tinkercad.com/things/0VyhmJc4OoO-amazing-esboo-snicket/editel?sharecode=00RdnXTjmQDQ\\_9fmgaSxejO8H1O77AOIXshQHfMY](https://www.tinkercad.com/things/0VyhmJc4OoO-amazing-esboo-snicket/editel?sharecode=00RdnXTjmQDQ_9fmgaSxejO8H1O77AOIXshQHfMY)

▶ ⏪ 1 (Arduino Uno R3) ▼

```

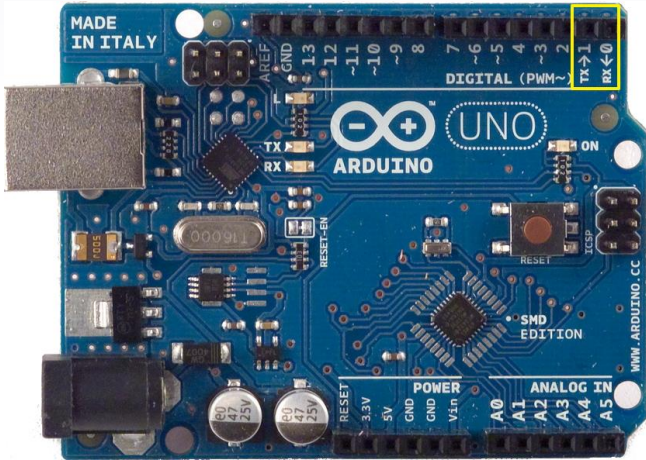
5
6
7 void setup() {
8   myServo.attach(8);
9   myServo.write(90);
10  delay(1000);
11  Serial.begin(9600);
12 }
13
14 void loop() {
15   cm = 0.01723 * readUltrasonicDistance(7, 6);
16   Serial.println(cm);
17   delay(500);
18   if (cm < 100) {
19     myServo.write(0);
20     delay(1000);
21   }
22   else {
23     myServo.write(90);
24     delay(1000);
25   }
26 }
27
28 long readUltrasonicDistance(int triggerPin, int echoPin)
29 {
30   pinMode(triggerPin, OUTPUT); // Clear the trigger
31   digitalWrite(triggerPin, LOW);
32   delayMicroseconds(2);
33   // Sets the trigger pin to HIGH state for 10 microseconds
34   digitalWrite(triggerPin, HIGH);
35   delayMicroseconds(10);
36   digitalWrite(triggerPin, LOW);
37   pinMode(echoPin, INPUT);
38   // Reads the echo pin, and returns the sound wave travel time in microseconds
39   return pulseIn(echoPin, HIGH);
40 }
41

```

📄 串列監視器

118	240
142	
100	
147	
145	
147	
121	
120	60

## (B) Arduino IDE



```
Blink | Arduino 1.8.5

This example code is in the public domain.

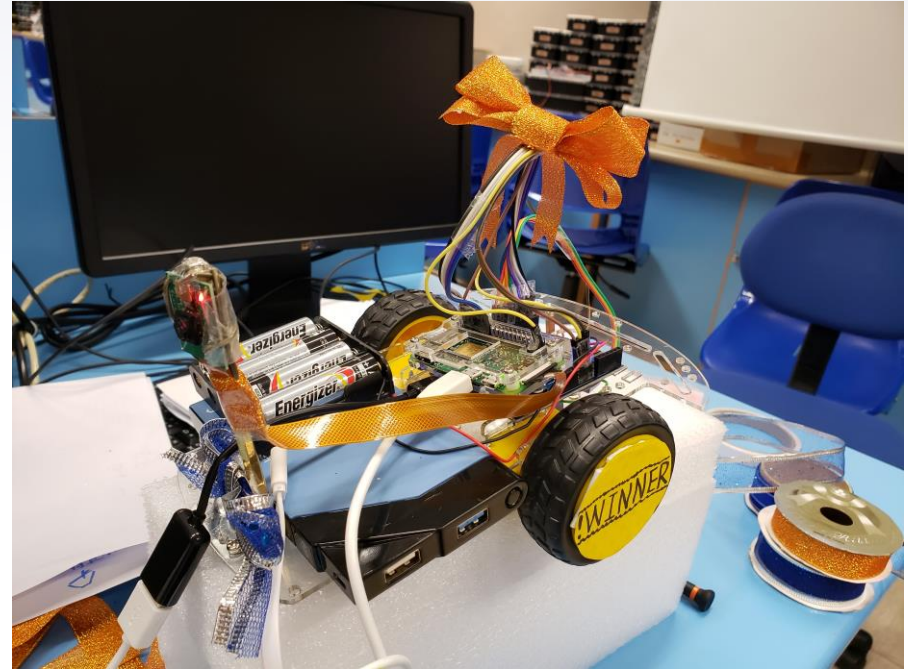
http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {$
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

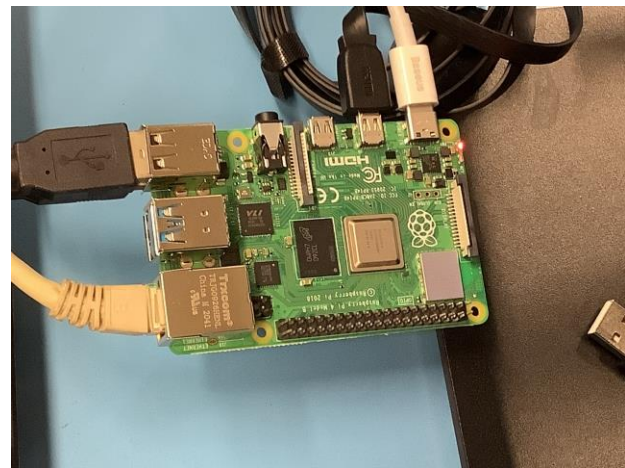


## (C) Raspberry Pi



左鏡頭拍攝的中心點

右鏡頭拍攝的中心點





# Conclusion: Event handler

- An **event** is an action that takes place when a user interacts with a program.
- Each event can be in a high level, e.g. GUI events include **key presses**, **mouse movement**, **action selections**, and **timers expiring**.
- On a lower level, events can represent **availability of new data** for reading a file or **network stream**.
- In programming, an **event handler** is a routine that **operates once an event takes place**.
- An event handler can be implemented in an **event loop** that **waits for events and dispatches events or messages** in a program.

[https://en.wikipedia.org/wiki/Event\\_\(computing\)](https://en.wikipedia.org/wiki/Event_(computing))

[https://en.wikipedia.org/wiki/Event\\_loop](https://en.wikipedia.org/wiki/Event_loop)

<https://www.technologyuk.net/computing/software-development/software-design/event-driven-programming.shtml>

Thank you!!

The Next Session will begin!

