



高一資訊科技銜接課程

授課教師：余淙玉

為何要上銜接課程？協助剛入學的新生順利銜接新舊課綱課程

銜接課程分為教師授課的**實體課程**，和學生上網學習的**線上課程**兩種方式。

- 1. 資訊科技**：為108年新課綱實施所新增之領域課程，為弭平新舊課綱學習內容之落差，規劃銜接課程內容包含語言2節、演算法5節及程式設計9節，共計16節；本校規劃6節實體課程於升高一暑假期間上課，10節線上課程，請同學利用暑假時間自行上網學習。
- 2. 生物**：銜接需求共有免疫與演替兩個單元，免疫單元針對高一新生傳達免疫的概念與預防注射的重要性，以補足學習空缺，演替單元以影片或投影片介紹演替的過程，讓學生了解演替的意義與在環境經營上的應用。由於這兩單元對於升高一學生而言，非常容易理解及學習，因此以線上課程方式，讓學生自行上網修習相關內容。
- 3. 化學**：在新課綱中，特別參照學生的認知發展歷程，將化學由巨觀到微觀的發展歷程，重新調整安排至各學習階段中，強調學習內容的連貫，並改善內容重複的問題；此外，亦簡化7~9年級的化學計量部分，改以實作課程加深學生對觀念的理解。為因應此調整，學生修習化學時，需將部分重要的概念重新說明與加強，並針對**實作課程**的原理說明補充，以利新、舊課綱的銜接。除了有影片可上網學習，亦規劃2節實體課程，於升高一暑假期間上課。

新課綱銜接教材數位平台

<https://ecc.pro.edu.tw>



教育部國民及學前教育署
K-12 Education Administration, Ministry of Education
新課綱銜接教材數位平臺



銜接教材



課程評量



最新消息



常見問題



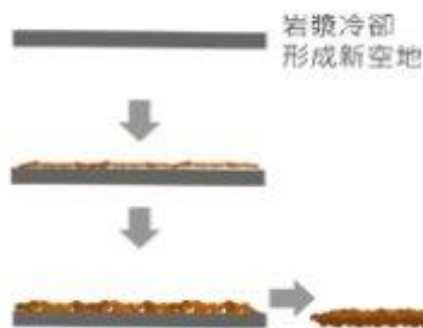
相關連結



下載專區



問題單反映



初級演替

演替的類型



教育部國民及學前教育署
K-12 Education Administration, Ministry of Education

國教署

化學學科中心

化學學科中心

生物學科中心

生物學科中心

資訊科技學科中心

資訊科技學科中心



新課綱銜接教材數位平台操作介紹

課程觀看管理

課程評量管理

銜接教材

🏠 首頁 > 銜接教材

選擇科目



資訊科技



生物



化學

新課綱銜接教材數位平台操作介紹－課程觀看

課程觀看管理

銜接教材

課程評量管理

銜接教材觀看

🏠 首頁 > 課程觀看管理 > 銜接教材 > 銜接教材觀看



語言-Blockly

發佈日期：108.06.14



新課綱銜接教材數位平台操作介紹 – 課程影片

操作影片列表 觀看課程影片時，建議使用(IE 11以上, Firefox, Google Chrome, Safari)瀏覽器，謝謝

觀看列表



語言-App Inventor

影片長度：00:30:57
發佈日期：108.06.14

App Inventor 2



語言-Python

影片長度：00:26:18
發佈日期：108.06.14

『Python程式設計簡介』

課程影片：

- 語言-Blockly
- 語言-App Inventor
- 語言-Python
- 語言-C++
- A-IV-1演算法基本概念
- A-IV-2陣列資料結構的概念與應用
- A-IV-3基本演算法的介紹
- P-IV-1程式語言基本概念、功能及應用
- P-IV-2結構化程式設計

新課綱銜接教材數位平台操作介紹－課程評量

課程觀看管理

課程評量管理

課程評量

課程評量

🏠 首頁 > 課程評量管理 > 課程評量

化學

生物

資訊科技

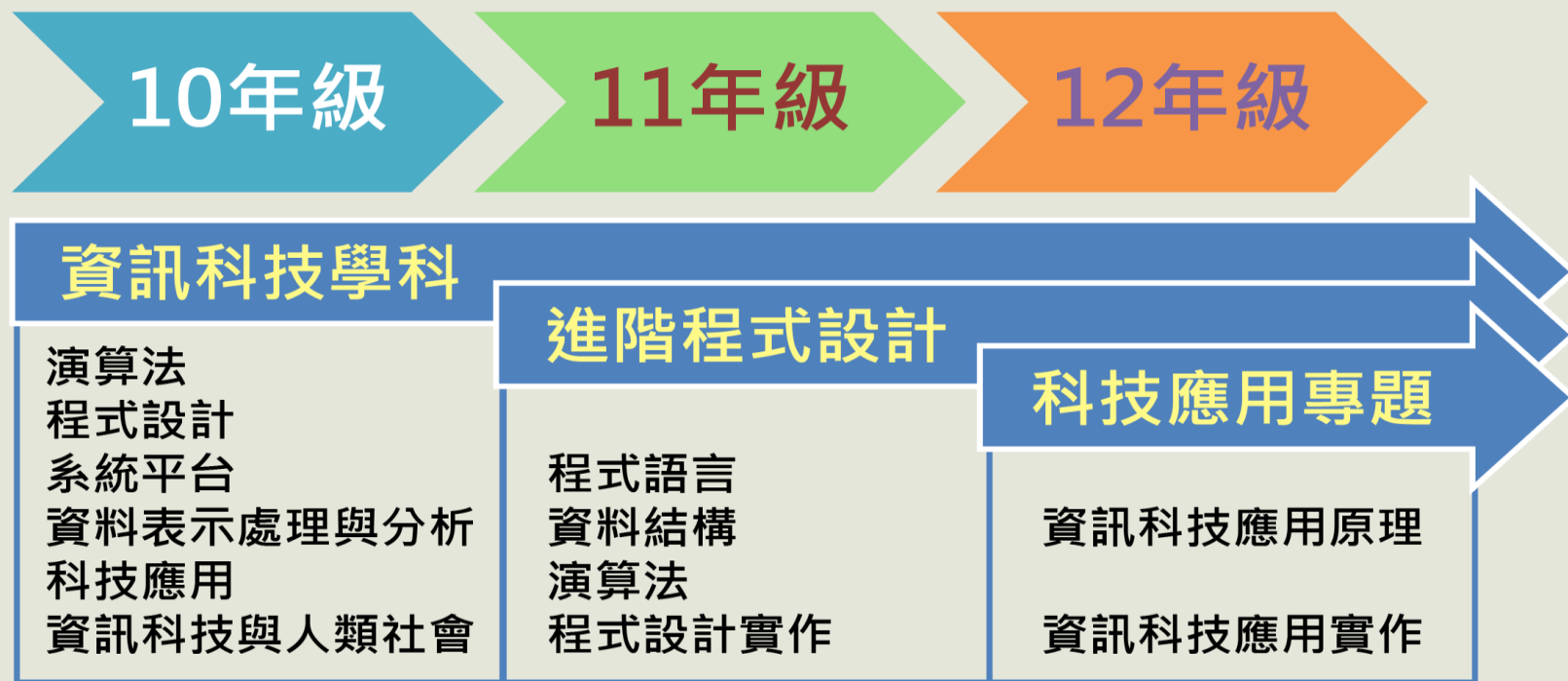
列表 ☰

※要至少完整看過一次影片，才能進行測驗!!

序	章節	測驗
1	語言-Blockly	進行測驗
2	語言-App Inventor	進行測驗

資訊科技課綱解讀與學科課程地圖發展(1/2)

資訊科技學科整體學習進程概覽圖



運算思維

資訊科技課綱解讀與學科課程地圖發展(2/2)



資訊科技 - 運算思維 Computational Thinking

運算思維：意指用電腦運算邏輯的概念來解決問題的思維模式。美國知名的電腦科學家 Jeannette M. Wing，曾經寫了一篇很有名的文章：[Computational Thinking](#)，文中開宗明義地提到「運算思維代表一個普遍適用的態度和技能，不僅僅是計算機科學家，任何人都將渴望學習和使用。」而 Google 也不遺餘力地推動CT教育，並提出了四個核心能力：

- 1. 拆解問題**：將一個任務或問題拆解成數個步驟或部分。
- 2. 找出規律性**：預測問題的規律，並找出模式做測試。
- 3. 歸納與抽象化**：找出最主要導致此模式的原則或因素。
- 4. 設計演算法**：設計出能夠解決類似問題並且能夠被重複執行的指令流程。

一個具有 CT 的人，可以運用既有的知識或工具，找出解決艱難問題的方法。你聽懂了嗎？

資訊科技 - 運算思維 Computational Thinking

運算思維，是與解決計算問題相關的思維過程(例如抽象、演算法設計、分解...等)和有形結果(例如自動化，數據表示，模式概括等)。其中包括和定義如下：[Google for Education](#)

- **抽象(Abstraction)**：識別和提取相關信息以定義主要構想。
- **演算法設計(Algorithm Design)**：創建一系列有序的指令，以解決類似問題或完成任務。
- **自動化(Automation)**：讓電腦或機器執行重複性的任務。
- **數據收集(Data Collection)**：收集信息。
- **數據分析(Data Analysis)**：通過發現模式或發展見解來使數據有意義。
- **數據表示(Data Representation)**：在適當的圖形、圖表、文字或圖像中描述和組織數據。
- **分解(Decomposition)**：將數據、流程或問題分解為較小的可管理部分。
- **並行化(Parallelization)**：從較大的任務同時處理較小的任務，以更有效地達到一個共同的目標。
- **模式概括(Pattern Generalization)**：創建觀察模式的模型、規則、原理或理論以測試預期結果。
- **模式識別(Pattern Recognition)**：觀察數據的模式、趨勢和規律性。
- **仿真(Simulation)**：開發模型來模擬現實世界的過程。

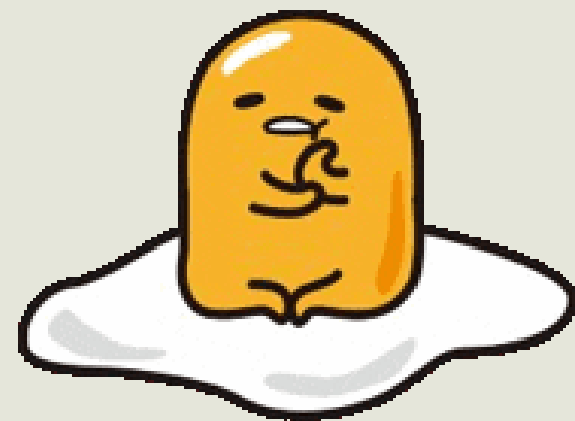
資訊科技 - 運算思維 Computational Thinking

動動腦：

假設在你的面前有兩個水壺，一個容量1000ml，另一個是600ml，如果要取得800ml的水，請問該如何做？

你的答案是...

我想想~



時間到！



動動腦：

假設在你的面前有兩個水壺，一個容量1000ml，另一個是600ml，如果要取得800ml的水，請問該如何做？

你的答案是...



資訊科技 – Blockly Games

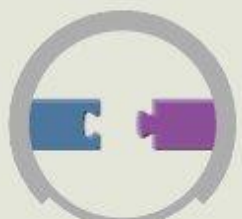
<https://blockly.games>

正體中文



為了明日的程式設計師所設計的遊戲。給教育者的資訊...

<https://blockly.games/>



拼圖



迷宮



小鳥



烏龜



影片



音樂



池塘遊戲教程



池塘

資訊科技 – Blockly Games

<https://blockly.games>

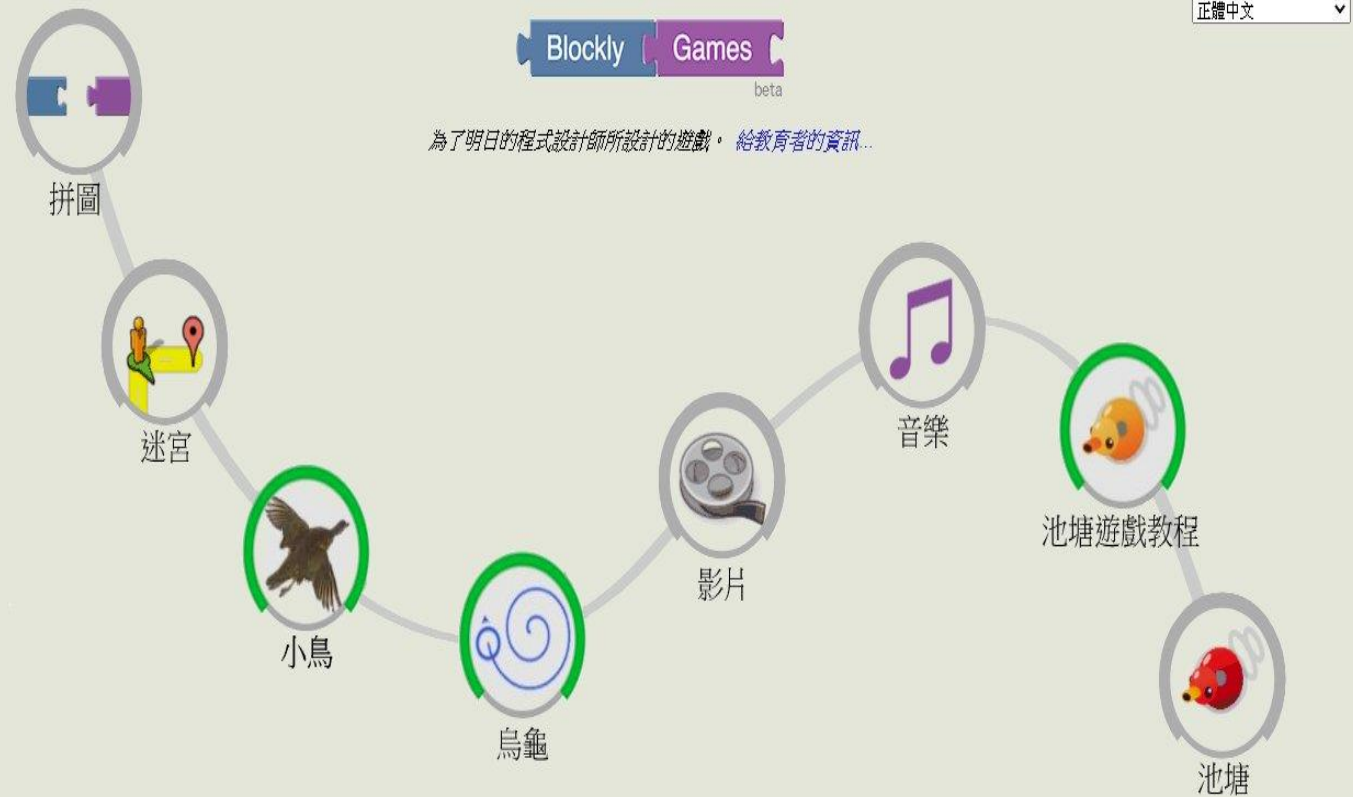
作業一：

請盡力完成「小鳥、烏龜、池塘遊戲教程」，完成後將首頁地圖截圖，檔名：[S10099_blockly.jpg](#)。將截圖檔寄到下列信箱：essa0517@gmail.com

信件主旨：

S10099的BlocklyGames作業

(S100→班級；99→座號)





程式設計基本概念

程式語言的基本要素



- **敘述(statement)**：

程式是由一行一行的敘述所組成的，每一個敘述必須符合程式語言的規則(語法)，才能正常執行。

- **變數(variable)**：

在程式執行的過程中，可能會改變內容的資料。必須按照程式語言的規範來命名及使用。

- **常數(constant)**：

在程式執行的過程中，不可以改變內容的資料。

$$A = \pi r^2$$

- **保留字(reserved word)**：

在程式語言中，具有特定意義的文字(例如函數及運算子)；不可以拼錯或做為其他用途。

- **註解(annotaion)**：

用來說明程式碼的文字，並不會被當作程式來執行。

程式設計的名詞解釋



- 準備輸入電腦去處理的東西稱作「資料(Data)」，而處理過後的結果稱為「資訊(Information)」
- **資料型態**：資料區分成許多不同的類型，例如：
 - 數值資料(可以計算的數值)
 - 字串資料(字元/文字)：字串的前後必須加上雙引號("...")
 - 日期資料(日期與時間)：日期時間的前後必須加上井字號(#...#)
 - 布林資料(條件判斷：真[True]、假[False]、且[and]、或[or]、否[not]...)：邏輯運算專用的資料
- **運算子、運算元與運算式**：
 - 運算子是用來運算資料項目的符號
 - 運算元是要運算的資料項目
 - 運算式是運算子和運算元的組合

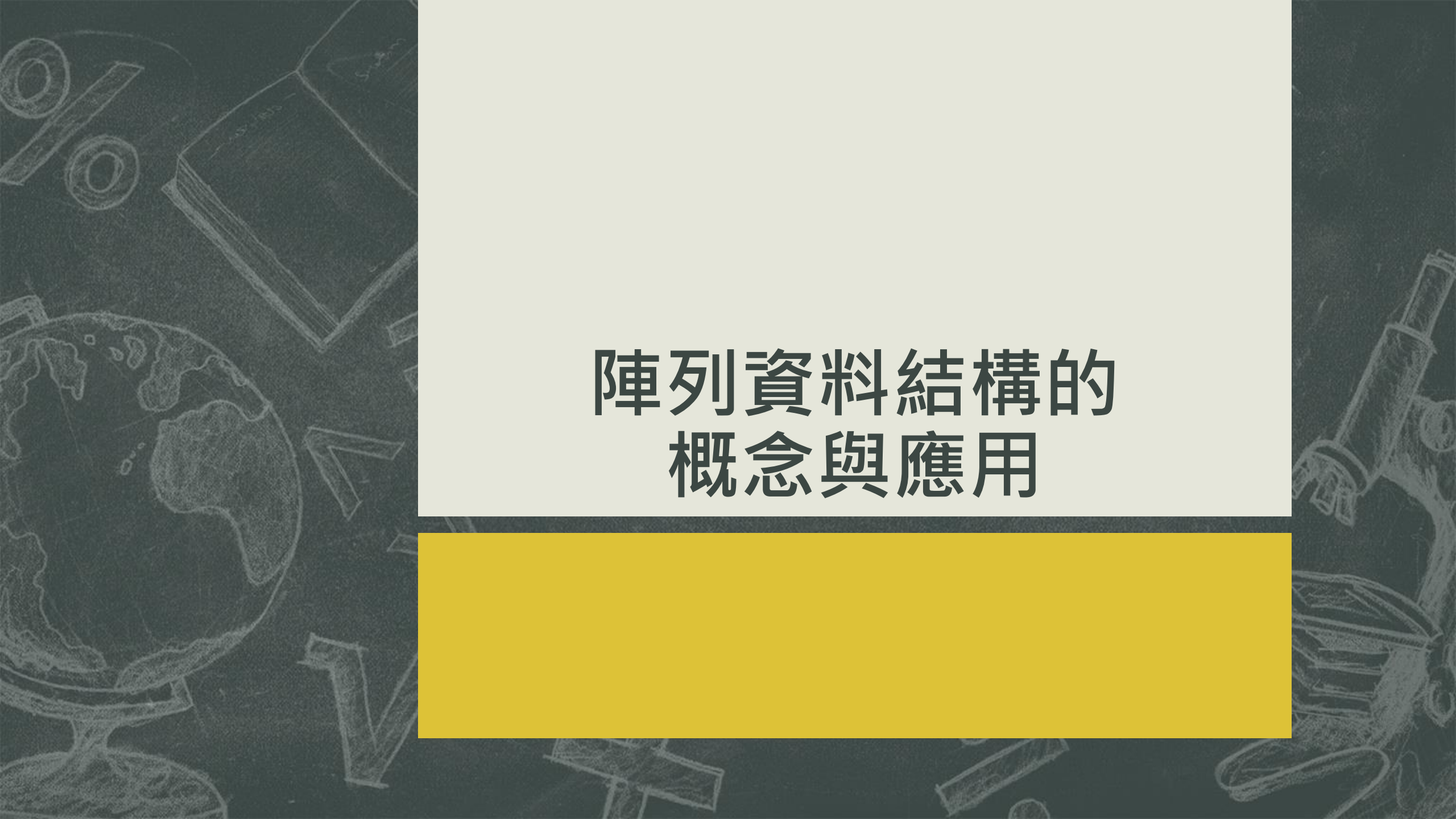
$$X = A + B$$

程式設計的名詞解釋



- **算術運算**：就是一般的數學運算。
+(加) -(減) *(乘) /(除) \ (整數除法取商)
mod(整數除法取餘數) ^(次方)
 $X = A + B$
- **比較運算**：比較兩邊的運算式。
=(等於), <(小於), >(大於), <>(不等於)
<=(小於等於), >=(大於等於)
- **邏輯運算**：邏輯判斷是非真假。
and(且) or(或) not(否)
- **連結運算**：連結兩個不同的資料。 + &
 $Z = X \& Y$ (例： $Z = \text{"abc"} + \text{"123"}$)

假設 $A = 2$ $B = 7$	$A > B$	False(假)
	$A \leq B$	True(真)
	$A < 1 \text{ and } B > 6$	False(假)
	$A > 1 \text{ and } B < 6$	False(假)
	$A < 1 \text{ or } B > 6$	True(真)
	$A > 1 \text{ or } B < 6$	True(真)
	Not ($A < B$)	False(假)



陣列資料結構的 概念與應用

變數 v.s. 陣列

■ Variables (變數)

- 讓電腦記住資料 (data)
- 可變的數，也就是其值可以隨著程式的執行過程而改變。
- 程式設計上的變數，代表的就是一個擁有名稱的記憶體儲存空間。

■ Array (陣列)

- 具有一個共同的名稱
- 使用連續記憶體空間
- 將一群形式相同的變數名稱，給予不同的數字編號做為註標(或稱為索引)
- **陣列簡單來說：就是具有相同變數名稱，但有不同註標的變數**

變數 v.s. 陣列

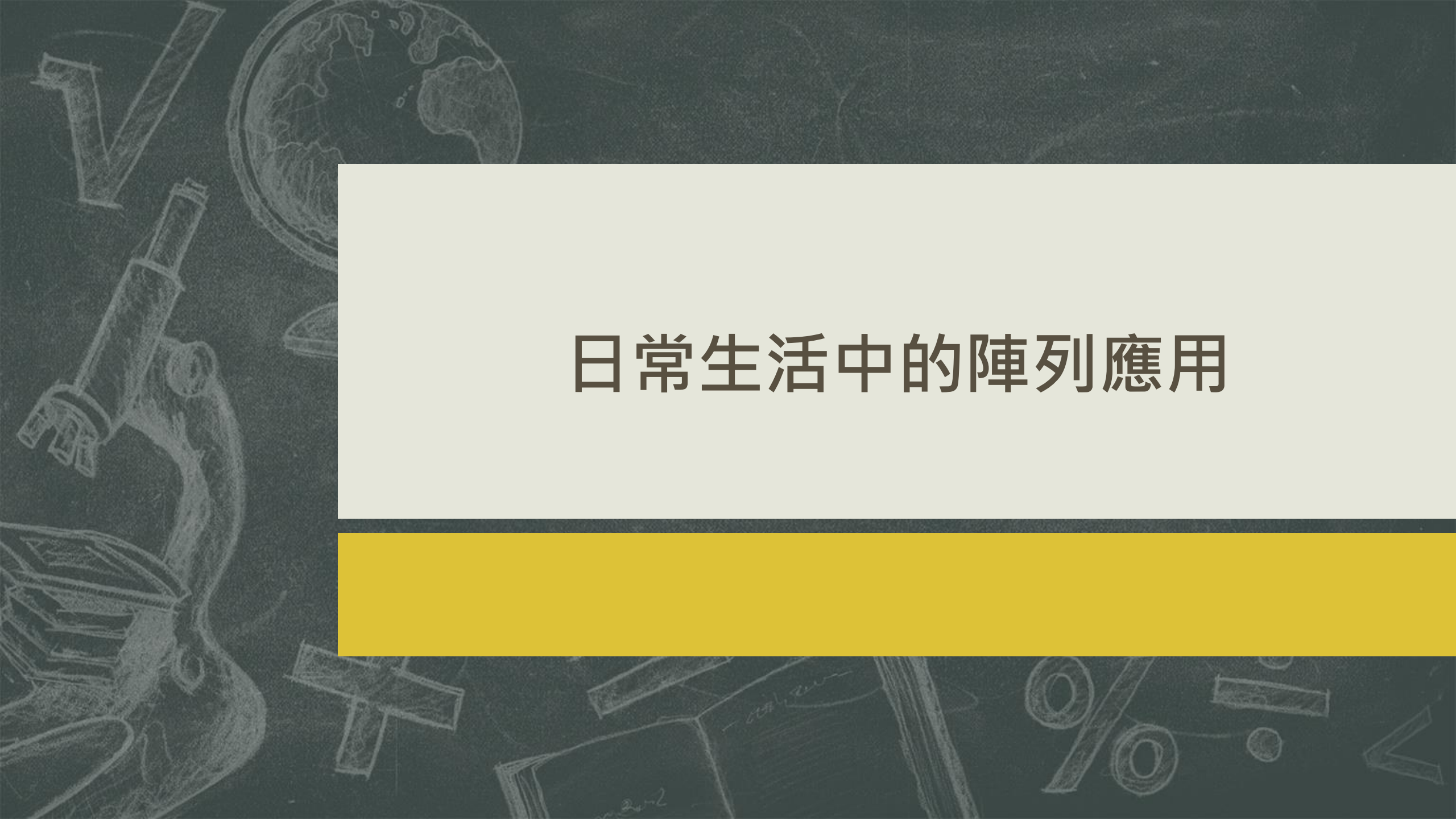
▪ Variables (變數)



▪ Array (陣列)



請問陣列(2)是哪種動物？



日常生活中的陣列應用

日常生活的陣列應用 – 火車

- 想像火車是一組經過編號的變數，如果每個變數相當於一個車廂號碼的話，陣列就可以是一列火車。
- 火車中的每節車廂我們稱之為索引，以便分辨這是哪一個車廂。
- 我們可以用：自強109[1]代表自強號火車109車次的第一節車廂，自強109[3]代表自強號火車109車次的第3個車廂。而「自強109」當成陣列的名稱，其中數字1,3就是陣列的索引值。



第一節車廂 火車頭，相當於變數名稱

- 陣列就如同變數一樣，可以儲存資料，也可以利用索引值來使用資料。
- 例如：台鐵可以記錄每節車廂剩餘的票數，那麼自強109 [1]=6，就可以代表自強109車次的第一節車廂剩餘6張車票。
- 自強109 [1]+自強109 [2]，就可以計算出自強109車次第一節車廂和第二節車廂剩餘的車票數。

陣列之應用、多維陣列

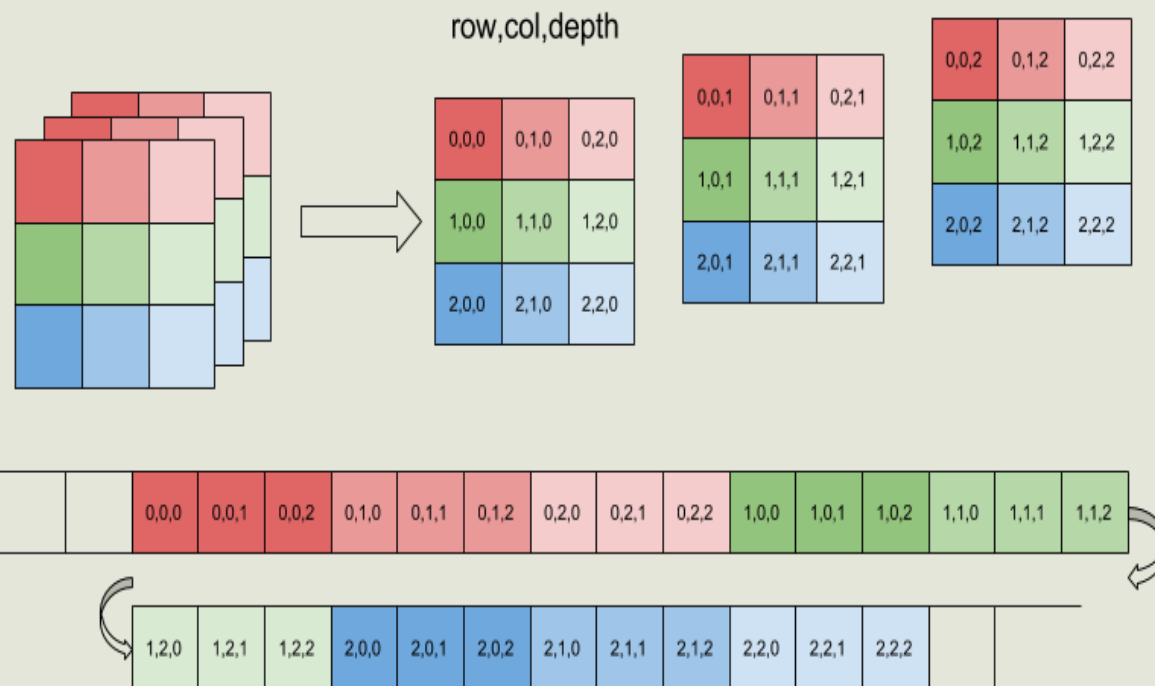
一維陣列：只紀錄單項資料，例如只紀錄同學的姓名。

姓名	小明	亞培	小智	小夫	大雄	浩浩
索引值	0	1	2	3	4	5

二維陣列：可同時紀錄多項資料，橫向為行、縱向為列。

姓名	小明	亞培	小智	小夫	大雄	浩浩
國文	95	80	95	60	50	75
數學	65	90	85	50	70	100
英文	75	60	90	55	60	90

三維陣列：多個二維陣列所組成。橫向為行、縱向為列、前後為頁。



生活中的陣列應用學習單

作業二：

- 在日常生活中，請舉出一個可應用陣列的實例。
- 定義陣列的名稱
- 定義陣列的索引值範圍
- 實際應用狀況
- 利用文書編輯軟體，將上述項目內容編輯成文書檔後，再寄到 essa0517@gmail.com

信件主旨：S10099的陣列應用作業

(S100→班級；99→座號)

統一發票號碼獎中獎號碼 | 雲端發票專屬獎中獎號碼

109年05-06月

類別	中獎號碼
特別獎	03016191 同期統一發票收執聯8位數號碼與特別獎號碼相同者獎金1,000萬元
特獎	62474899 同期統一發票收執聯8位數號碼與特獎號碼相同者獎金200萬元
頭獎	33657726 06142620 66429962 同期統一發票收執聯8位數號碼與頭獎號碼相同者獎金20萬元
二獎	同期統一發票收執聯末7位數號碼與頭獎中獎號碼末7位相同者各得獎金4萬元
三獎	同期統一發票收執聯末6位數號碼與頭獎中獎號碼末6位相同者各得獎金1萬元
四獎	同期統一發票收執聯末5位數號碼與頭獎中獎號碼末5位相同者各得獎金4千元
五獎	同期統一發票收執聯末4位數號碼與頭獎中獎號碼末4位相同者各得獎金1千元
六獎	同期統一發票收執聯末3位數號碼與頭獎中獎號碼末3位相同者各得獎金2百元
增開六獎	790 同期統一發票收執聯末3位數號碼與增開六獎號碼相同者各得獎金2百元



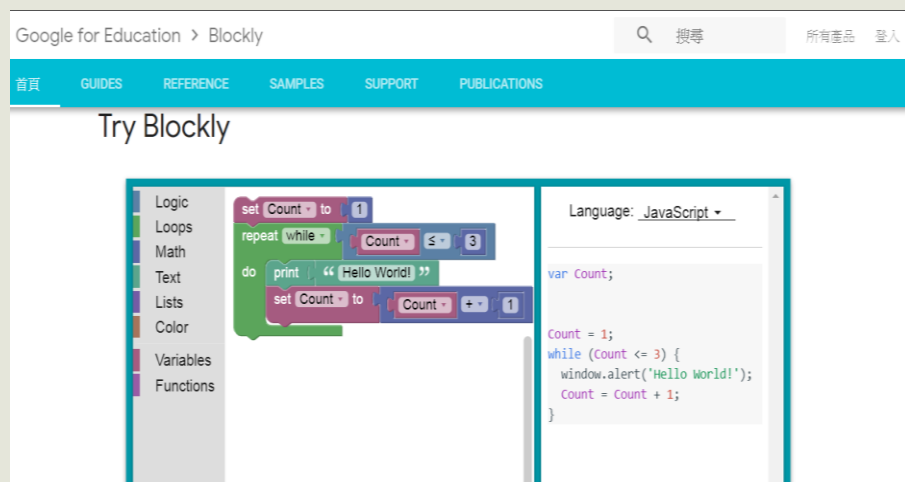
結構化程式設計 Blockly

Blockly 程式設計

- **Blockly** 是一款運行在網頁客戶端的 JavaScript 程式庫，目的是用來打造視覺化程式設計語言同時附帶編輯器。Blockly使用可在網頁上拖拉並相互連接在一起的視覺化方塊，使得程式碼的撰寫更加容易，並且可以產生JavaScript, Python, PHP ,Lua或 Dart等的代碼。

線上Blockly

<https://developers.google.com/blockly>



離線Blockly

- 下載離線版Blockly壓縮包：<https://bit.ly/2CulbYh>
- 下載後解壓縮
- 執行解壓縮後的Blockly.html，即可開始程式設計



Blockly 程式設計



blockly demos



全部

影片

新聞

圖片

地圖

更多

設定

工具

約有 14,400 項結果 (搜尋時間：0.28 秒)

blockly-demo.appspot.com > static > demos > code

✓ [程式碼 - Blockly Demo](#)

Blockly > Demos > 程式碼. Bahasa Melayu, Brezhoneg, Català, Dansk, Deutsch, Eesti, English, Español, Français, Hunsrik, Interlingua, Italiano, Kabyle, Lenga ...

✓ [Blockly Demo: Code](#)

Blockly > Demos > Code. Bahasa Melayu, Brezhoneg, Català ...

✓ [Blockly > Demos > Code](#)

Blockly > Demos > Code. Bahasa Melayu, Brezhoneg, Català ...

✓ [Blockly > Demos > Kod](#)

Blockly > Demos > Kod. Bahasa Melayu, Brezhoneg, Català ...

✓ [Blockly > Demos > Codi](#)

Blockly > Demos > Codi. Bahasa Melayu, Brezhoneg, Català ...

[blockly-demo.appspot.com 的其他相關資訊](#) »

Blockly 程式設計

Blockly > Demos > 程式碼

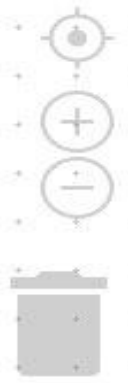
正體中文

積木	JavaScript	Python	PHP	Lua	Dart	XML
邏輯	邏輯	邏輯	邏輯	邏輯	邏輯	邏輯
迴圈	迴圈	迴圈	迴圈	迴圈	迴圈	迴圈
數學式	數學式	數學式	數學式	數學式	數學式	數學式
文字	文字	文字	文字	文字	文字	文字
列表	列表	列表	列表	列表	列表	列表
顏色	顏色	顏色	顏色	顏色	顏色	顏色
變量	變量	變量	變量	變量	變量	變量
流程	流程	流程	流程	流程	流程	流程



Blockly code blocks:

- 到 做些什麼 (To do something)
- 到 做些什麼 (To do something)
- 返回 (Return)
- 如果 返回 (If return)



Blockly 程式設計 – 簡單範例

Blockly > Demos > 程式碼

正體中文

積木

JavaScript

Python

PHP

Lua

Dart

XML

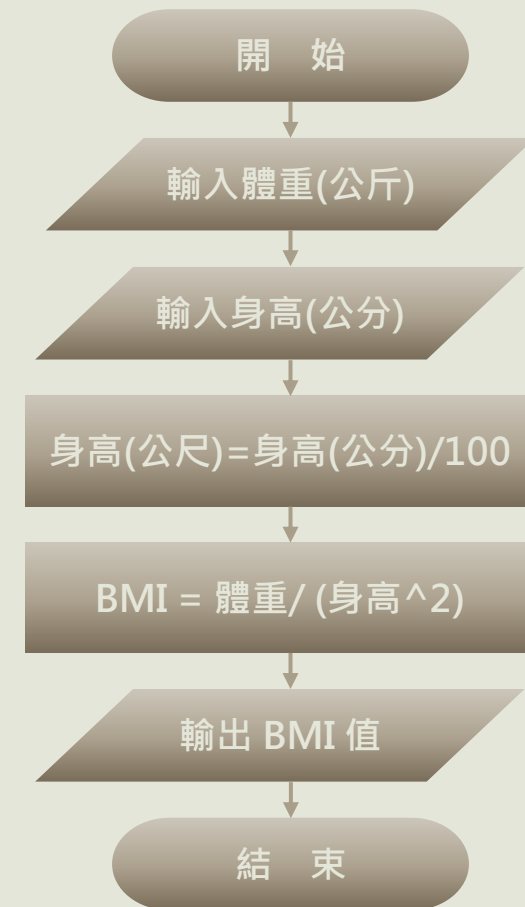


GO



```
<xml xmlns="https://developers.google.com/blockly/xml">
  <variables>
    <variable id="s@3k6G3WDgbpyu|IN8yV">a</variable>
    <variable id="ff(,tIxflDGR;d[*@Uq$">b</variable>
  </variables>
  <block type="variables_set" id="]k,N$V)qygh)x^`=ms?[" x="10" y="10">
    <field name="VAR" id="s@3k6G3WDgbpyu|IN8yV">a</field>
    <value name="VALUE">
      <block type="math_number" id="F+{U;/dESr;@A*ECTATr">
        <field name="NUM">1</field>
      </block>
    </value>
    <next>
      <block type="variables_set" id="=!O!Aaa8@XJ@NRN/vAh[">
        <field name="VAR" id="ff(,tIxflDGR;d[*@Uq$">b</field>
        <value name="VALUE">
          <block type="math_number" id="F6[*e$pN8$^]^X5u*Q#D">
            <field name="NUM">2</field>
          </block>
        </value>
        <next>
          <block type="text_print" id="~q}T_6DAY*rE_Lvmsh1e">
            <value name="TEXT">
              <shadow xmlns="http://www.w3.org/1999/xhtml" type="text" id=">
                <field name="TEXT">abc</field>
              </shadow>
            </value>
            <block type="math_arithmetic" id="^Y$6Iy.V::+dR%JITc1{">
```


程式設計 – 循序結構及實例



計算BMI值程式實例：

- 設計一程式由使用者輸入身高及體重，計算出BMI值並顯示BMI值。
- $BMI = \text{體重(公斤)} / \text{身高(公尺)}^2$

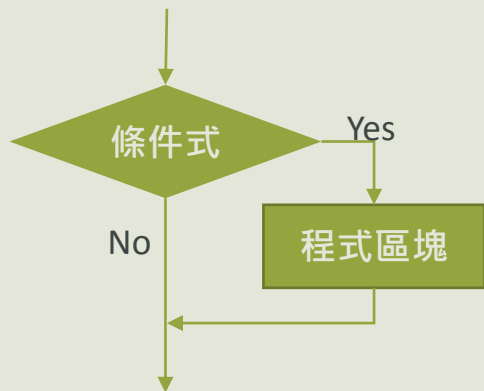
程式設計 – 選擇結構

單一選擇結構：

If...Then...End IF

If 條件式 Then 單一敘述 End If

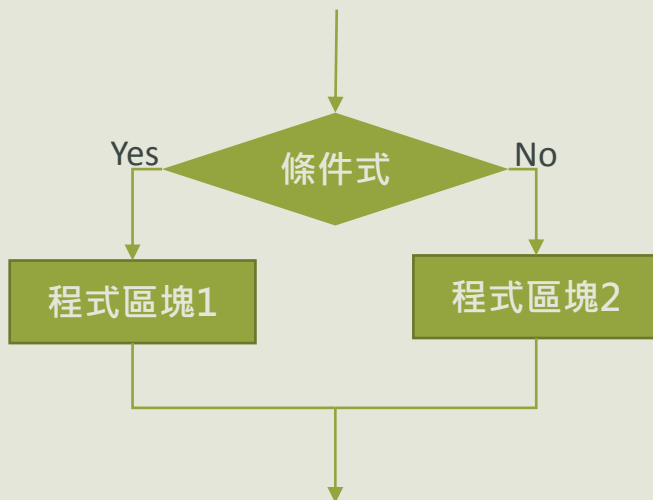
If 條件式 Then
... (程式區塊)
End If



雙重選擇結構：

If...Then...Else...End IF

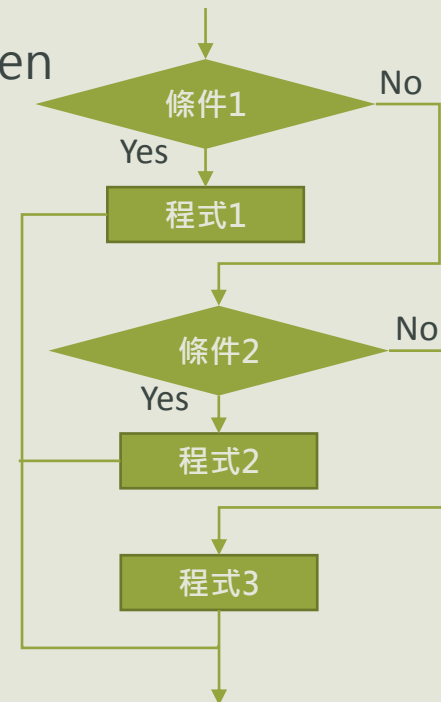
If 條件式 Then
... (程式區塊1)
Else
... (程式區塊2)
End If



多重選擇結構：

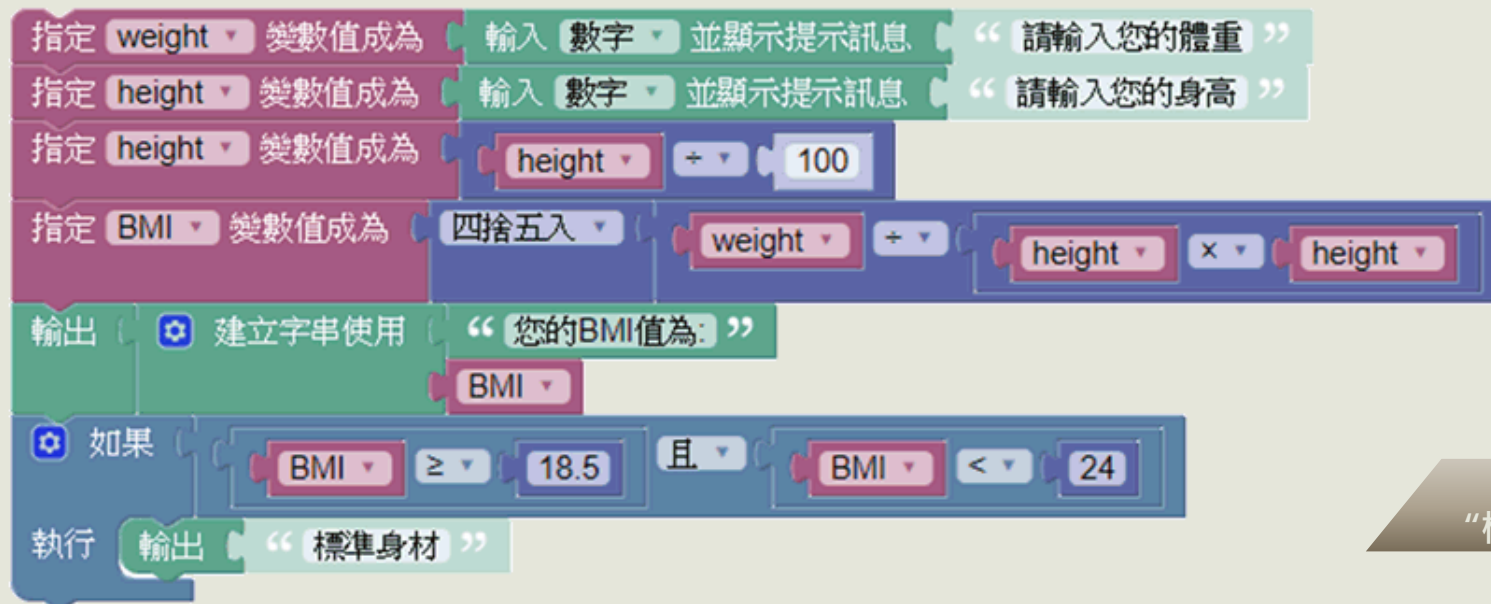
If...Then...ElseIf...Then...Else...End IF

If 條件式1 Then
... (程式區塊1)
ElseIf 條件式2 Then
... (程式區塊2)
Else
... (程式區塊3)
End If



程式設計 – 單一選擇結構實例

- 以單一選擇結構計算BMI值
- 當 $18.5 \leq \text{BMI} < 24$ 時顯示"標準身材"



程式設計 – 雙重及多重選擇結構實例

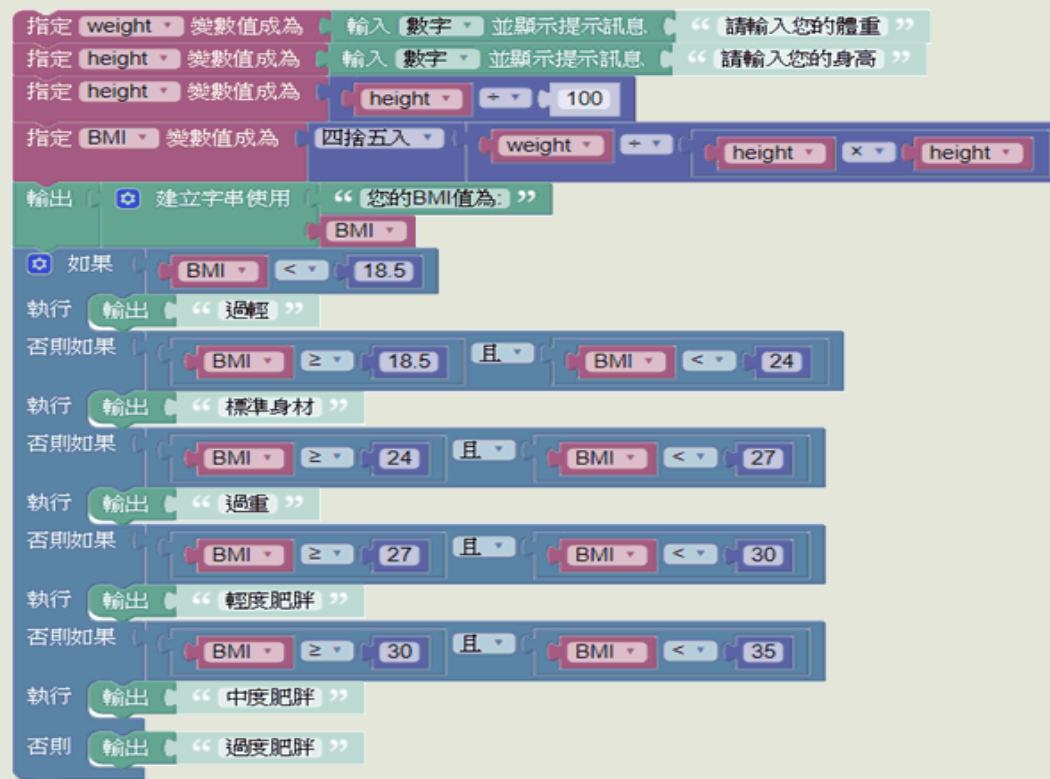
■ 以雙重選擇結構計算BMI值

當 $18.5 \leq \text{BMI} < 24$ 時顯示"標準身材"，
否則顯示 " 體重不標準"



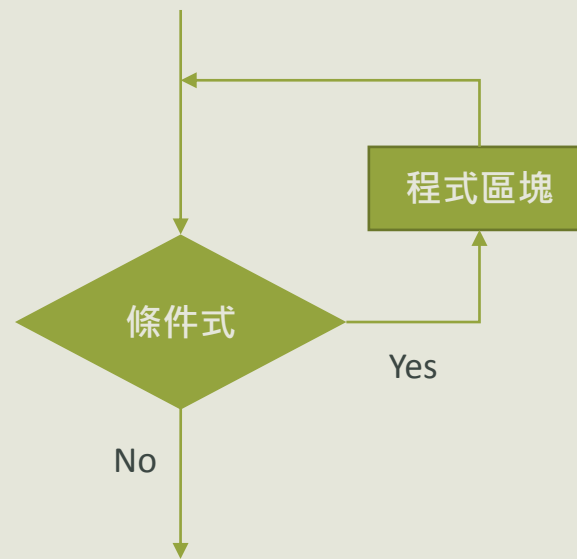
■ 以多重結構計算BMI值

- (a) $\text{BMI} < 18.5$ 輸出：過輕
- (b) $18.5 \leq \text{BMI} < 24$ 輸出：標準身材
- (c) $24 \leq \text{BMI} < 27$ 輸出：過重
- (d) $27 \leq \text{BMI} < 30$ 輸出：輕度肥胖
- (e) $30 \leq \text{BMI} < 35$ 輸出：中度肥胖
- (f) $\text{BMI} \geq 35$ 輸出：重度肥胖



程式設計 – 重覆結構及實例

- 使用變數控制迴圈執行的次數
- 一開始指定一個初始值給變數，若變數值小於等於終值，則執行迴圈內的敘述
- 再使用增值更新變數的值，並檢查是否小於等於終值，如此重複執行，直到變數值大於終值，才離開迴圈
- 設計解決累加問 $1+2+3+4+\dots+98+99+100=?$

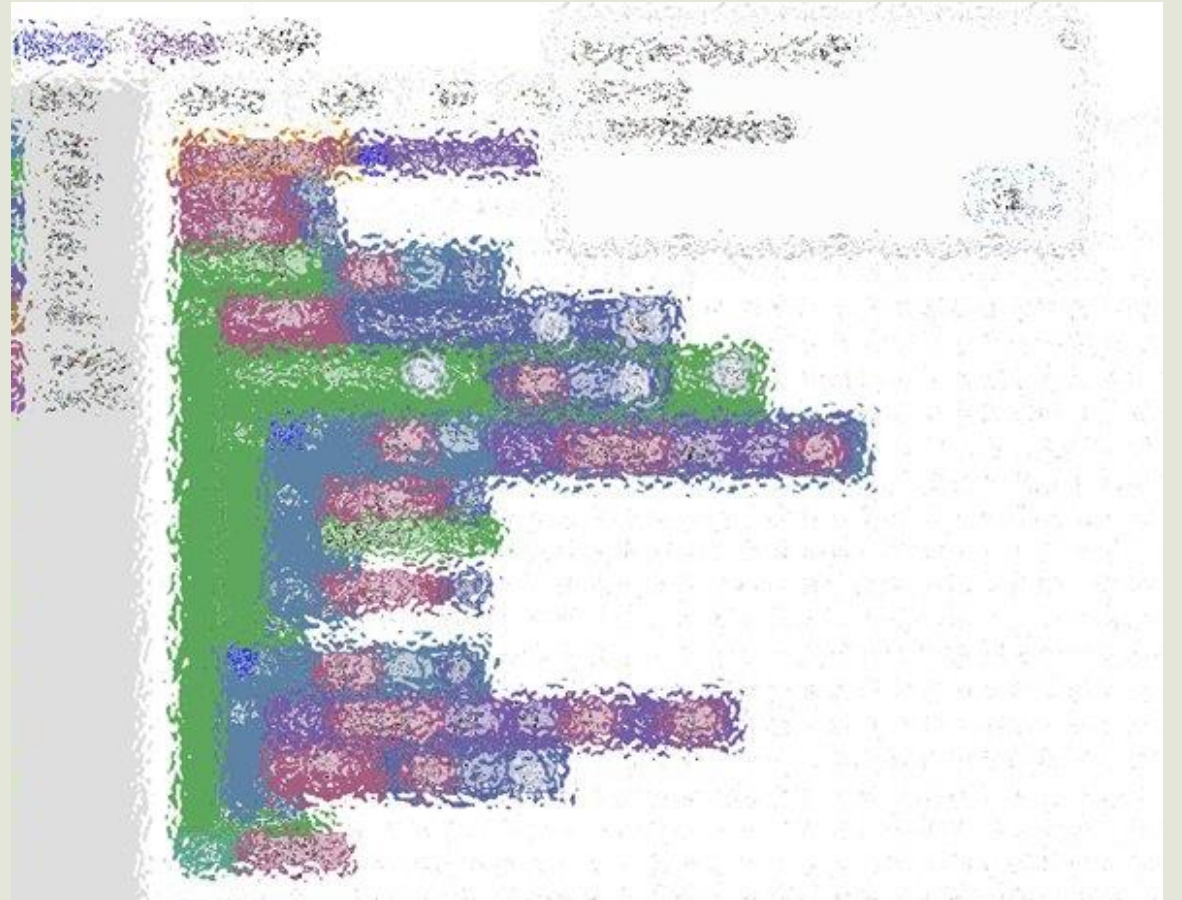


生活中的應用程式實作

作業三：

- 請使用 Blockly 寫出下列程式：
 1. 溫度轉換（華氏 \leftrightarrow 攝氏）程式
 2. 一元二次方程式求解
 3. 電腦隨機選取6個1~42之間不重複數字
- 請繳交完成作答畫面截圖。
- 檔名：[S10099_程式名稱.jpg](#)，再寄到 essa0517@gmail.com

信件主旨：[S10099](#)的應用程式實作
([S100](#)→班級；[99](#)→座號)



臺北市立萬芳高級中學

一所自主有愛、迎向世界的學校



自主學習

自我管理

藝術涵養

多元探索



同理有愛

溝通表達

協同合作

人文關懷



接軌世界

系統思考

全球思維

創新應變