

# 110新泰創造力資優

2021-1113  
趙和修

# 數學計算-說明畫面

複製Ctrl+C，貼上Ctrl+V

主角說出：這是1個加法練習程式，按下任何鍵開始。

手機、平板

電腦(鍵盤)

與 鼠標 的間距

事件

詢問 你的名字是? 並等待

詢問 這是一個加法練習遊戲，按下任何鍵開始 並等待

詢問的答案

任何 鍵被按下?

滑鼠鍵被按下?

鼠標的 x

鼠標的 y

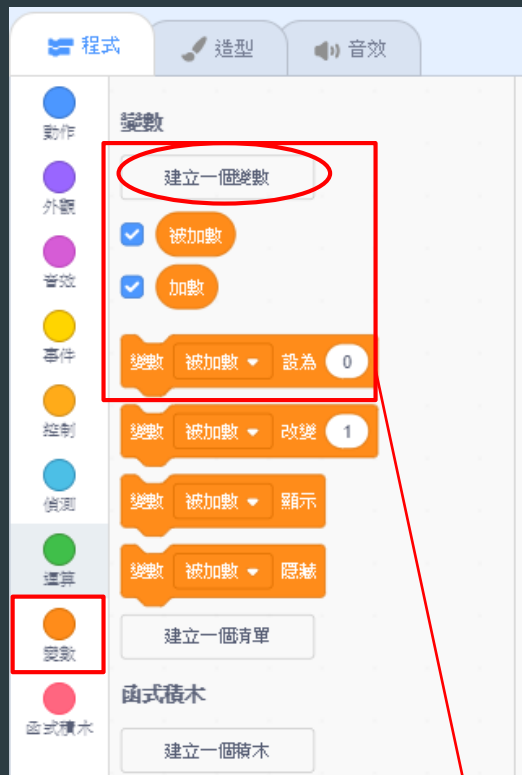
當 被點擊

當 被點擊

說出 這是一個加法練習遊戲，按下任何鍵開始

等待直到 任何 鍵被按下?

# 建立2個變數(例如：被加數、加數)



# 字串組合

● 事件

● 控制

● 偵測

● 字串組合

● 變數

- 且
- 或
- 不成立
- 字串組合 apple banana
- 字串 apple 的第 1 字
- 字串 apple 的長度

● 事件

● 控制

● 偵測

● 變數

● 區式積木

- 顏色 碰到 顏色 ?
- 與 鼠標 的間距
- 詢問 What's your name? 並等待
- 詢問的答案
- 空白 鍵被按下?
- 滑鼠鍵被按下?
- 鼠標的 x

字串組合 apple banana

字串組合 apple banana

字串組合 apple 字串組合 apple 字串組合 apple 字串組合 apple banana

字串組合 請問 字串組合 被加數 字串組合 + 字串組合 加數 =?

詢問 字串組合 請問 字串組合 被加數 字串組合 + 字串組合 加數 =? 並等待

# 程式積木

Scratch block palette categories and examples:

- 動作 (Action): 重複無限次 (Repeat Forever)
- 外觀 (Appearance): 如果...那麼 (If...Then)
- 音效 (Sound): 如果...那麼 (If...Then)
- 事件 (Events): 如果...那麼 (If...Then)
- 控制 (Control): 如果...那麼 (If...Then), 否則 (Else)
- 偵測 (Motion): 等待直到 (Wait Until)
- 運算 (Math): 重複直到 (Repeat Until)
- 變數 (Variables): 變數 (Variables)
- 函式積木 (Functions): 函式積木 (Functions)

Scratch 'Math' block category:

- 運算 (Math):
  - 加 (+)
  - 減 (-)
  - 乘 (\*)
  - 除 (/)
  - 隨機取數 (Random Number) 1 到 10
  - 比較運算 (Comparison):
    - > 50
    - < 50
    - = 50

Scratch 'Query' block category:

- 詢問 (Query):
  - 碰到顏色? (Click on Color?)
  - 顏色 碰到顏色? (Color Click on Color?)
  - 與 鼠標 的間距 (Distance to Mouse)
  - 詢問 "What's your name?" 並等待 (Ask "What's your name?" and wait)
  - 詢問的答案 (Answer)
  - 空白 鍵被按下? (Blank Key Pressed?)
  - 滑鼠鍵被按下? (Mouse Key Pressed?)
  - 鼠標的 x (Mouse X)

Scratch 'Variables' block category:

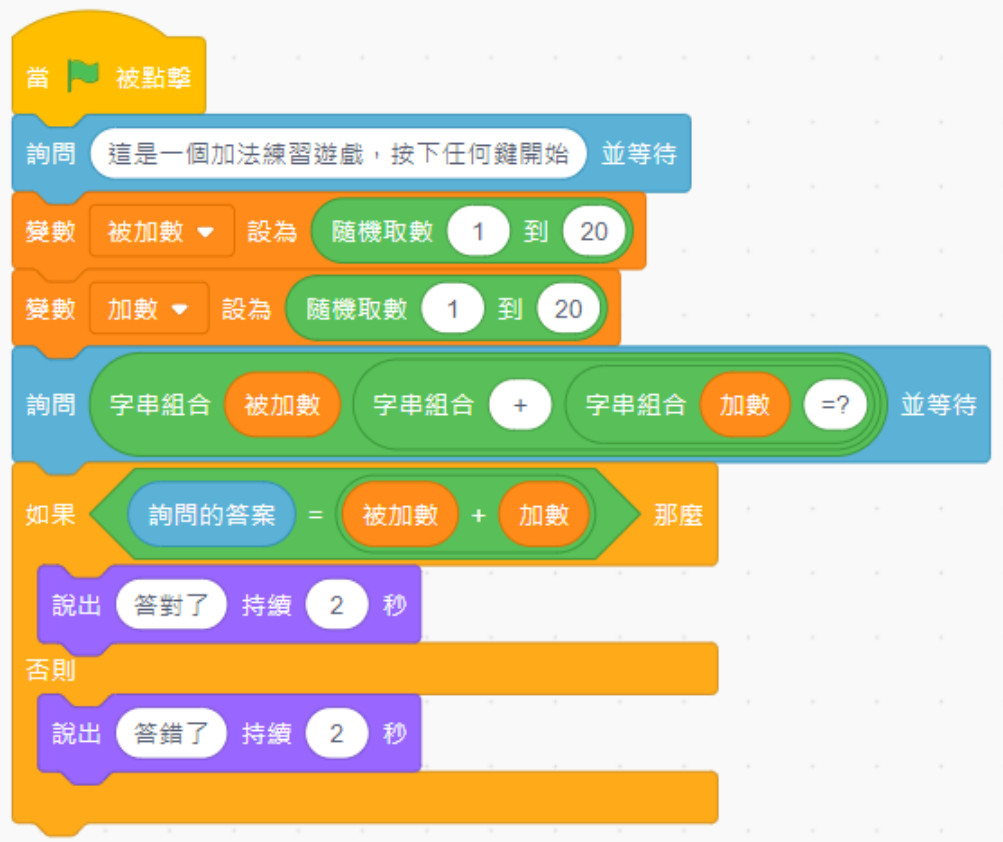
- 變數 (Variables):
  - 建立一個變數 (Create a Variable)
  - 被加數 (Addend)
  - 加數 (Addend)
  - 變數 被加數 設為 0 (Set Variable to 0)
  - 變數 被加數 改變 1 (Change Variable by 1)

Scratch script snippet:

```
如果 詢問的答案 = 被加數 + 加數 那麼  
否則
```

# 加法練習-1題

如果「詢問的答案」等於被加數+加數，就回答「答對了」，否則就回答「答錯了」。



# 加法練習-5題

增加重複5次

當 被點擊

詢問 這是一個加法練習遊戲，按下任何鍵開始 並等待

重複 5 次

變數 被加數 設為 隨機取數 1 到 20

變數 加數 設為 隨機取數 1 到 20

詢問 字串組合 被加數 字串組合 + 字串組合 加數 =? 並等待

如果 詢問的答案 = 被加數 + 加數 那麼

說出 答對了 持續 2 秒

否則

說出 答錯了 持續 2 秒

↻

Detailed description: The image shows a Scratch script for a math practice game. It starts with a 'When clicked' event, followed by a 'Say' block with the text '這是一個加法練習遊戲，按下任何鍵開始' and a 'Wait' block. A 'Repeat' block is highlighted with a red box and an arrow from the text '增加重複5次'. Inside the repeat block, there are two 'Set variable to' blocks: '被加數' (addend) and '加數' (addend), both set to 'Random number from 1 to 20'. This is followed by an 'Ask' block with the text '字串組合 被加數 字串組合 + 字串組合 加數 =?' and a 'Wait' block. Then, an 'If' block checks if '詢問的答案' (the answer) equals '被加數 + 加數'. If true, it says '答對了' (Correct) for 2 seconds; otherwise, it says '答錯了' (Incorrect) for 2 seconds. The script ends with a 'Repeat' block's arrow icon.

# 加法練習-答對幾題？

增加變數答對題數，

答對題數開始**設為0**

如果「**詢問的答案**」等於

**被加數+加數**，答對題數

**改變1**(增加1)。

最後說出答對幾題。

The image shows a Scratch script for a math quiz game. The script starts with a '當被點擊' (When clicked) event block. It then asks the user '這是一個加法練習遊戲，按下任意鍵開始' (This is a math practice game, press any key to start) and waits. A variable '答對題數' (Number of correct answers) is set to 0. A loop repeats 5 times. Inside the loop, two random numbers between 1 and 20 are generated for '被加數' (Addend) and '加數' (Addend). The user is asked to solve the addition problem. If the user's answer is correct, the '答對題數' variable is increased by 1. If the answer is wrong, a message is spoken. Finally, the total number of correct answers is spoken.

```
當被點擊
詢問 這是一個加法練習遊戲，按下任意鍵開始 並等待
變數 答對題數 設為 0
重複 5 次
  變數 被加數 設為 隨機取數 1 到 20
  變數 加數 設為 隨機取數 1 到 20
  詢問 字串組合 被加數 字串組合 + 字串組合 加數 =? 並等待
  如果 詢問的答案 = 被加數 + 加數 那麼
    變數 答對題數 改變 1
  說出 答對了 持續 2 秒
  否則
    說出 答錯了 持續 2 秒
  說出 字串組合 你總共答對 字串組合 答對題數 題。
```



# 修改

- ▶ 請將加法練習改成乘法練習-作業
- ▶ 請將加法練習改成減法練習  
(如何設計才不會有負的答案)
- ▶ 請將加法練習改成除法練習-作業  
(如何設計才不會有小數點的答案)

# 減法練習

被減數-減數=差  
被加數+加數=和

如何設計才不會有負的答案：

1. 被減數大於減數
2. 和-被加數=加數  
和-加數=被加數

# 除法練習

被乘數 $\times$ 乘數=積

被除數 $\div$ 除數=商加餘數

如何設計才不會有小數點的答案：

積 $\div$ 被乘數=乘數

積 $\div$ 乘數=被乘數

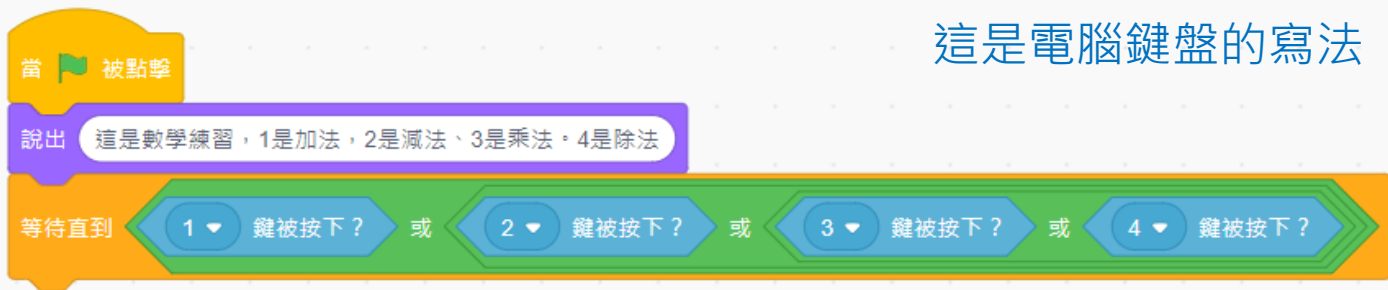
# 加減乘除法整合

主角開始說出：  
這是數學練習，1是加法，2是減法、  
3是乘法。4是除法

這是手機平板的寫法



這是電腦鍵盤的寫法



# 如何回到選單畫面(再玩1次)

增加1個選單的廣播訊息(背景)  
程式一開始便廣播選單



# 如何回到選單畫面(再玩1次)

原程式的最後的說出，改為詢問，加上  
”，按下R鍵回到選單，其他鍵結束。”



# 終極密碼-猜數字陽春版01

- ▶ 主角說：這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。
- ▶ 亂數取得“終極密碼”
- ▶ 遊戲開始  
主角提問，等待回答，  
如果“回答”小於“終極密碼”，說“太小了”，廣播遊戲開始。  
如果“回答”大於“終極密碼”，說“太大了”，廣播遊戲開始。  
如果“回答”等於“終極密碼”，說“答對了”，遊戲結束。

# 陽春版01： 電腦操作且不考慮再玩1次

## A1寫法

The A1 version code starts with a '當被點擊' (When clicked) event. It sets the stage to 'x: -120 y: -20', says '這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。', and waits for '任何鍵被按下?' (Any key pressed?). It then generates a random number between 1 and 99 for '終極密碼' (Final password) and broadcasts '開始猜' (Start guessing). A loop of '詢問' (Ask) and '如果' (If) blocks follows. The '如果' blocks check if the answer is greater than, less than, or equal to the password, providing feedback like '太大了' (Too big) or '太小了' (Too small) with a 2-second wait, and broadcasting '開始猜' again. The loop ends with '如果' (If) the answer is correct, displaying the password and saying '答對了' (Correct) with a 2-second wait.

電腦操作，  
按空白鍵很方便

## B1寫法

The B1 version code starts with a '當被點擊' (When clicked) event. It sets the stage to 'x: -120 y: -20', says '這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。', and waits for '任何鍵被按下?' (Any key pressed?). It then generates a random number between 1 and 99 for '終極密碼' (Final password) and repeats until the '詢問' (Ask) block returns the correct answer. The '如果' (If) blocks check if the answer is greater than, less than, or equal to the password, providing feedback like '太大了' (Too big) or '太小了' (Too small) with a 2-second wait. The loop ends with '如果' (If) the answer is correct, displaying the password and saying '答對了' (Correct) with a 2-second wait.



# 陽春版02：平板可操作

- ▶ 增加1個廣播訊息，例如**說明**(或變數設定)，最後再玩1次用
- ▶ 原**說出**改為**詢問**，給手機平板用。

### A2寫法

```
當 被點擊  
定位到 x: -120 y: -20  
廣播訊息 說明  
當收到訊息 說明  
詢問 這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。 並等待  
變數 終極密碼 隱藏  
變數 終極密碼 設為 隨機取數 1 到 99  
廣播訊息 開始猜  
如果 詢問的答案 > 終極密碼 那麼  
說出 太大了 持續 2 秒  
廣播訊息 開始猜  
如果 詢問的答案 < 終極密碼 那麼  
說出 太小了 持續 2 秒  
廣播訊息 開始猜  
重複直到 詢問的答案 = 終極密碼  
變數 終極密碼 顯示  
說出 答對了 持續 2 秒
```

### B2寫法

```
當收到訊息 說明  
定位到 x: -120 y: -20  
廣播訊息 說明  
詢問 這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。 並等待  
變數 終極密碼 隱藏  
變數 終極密碼 設為 隨機取數 1 到 99  
重複直到 詢問的答案 = 終極密碼  
詢問 請問是終極密碼是多少？ 並等待  
如果 詢問的答案 > 終極密碼 那麼  
說出 太大了 持續 2 秒  
廣播訊息 開始猜  
如果 詢問的答案 < 終極密碼 那麼  
說出 太小了 持續 2 秒  
廣播訊息 開始猜  
變數 終極密碼 顯示  
說出 答對了 持續 2 秒
```

# 陽春版03：增加猜答次數

- ▶ 增加一個變數例如**次數**，紀錄主角答題次數。
- ▶ 一開始**“次數”**設為0次，每次回答後**“次數”**改變1 (增加1次)。
- ▶ 最後組合字串



**B3寫法**

# 陽春版03：增加猜答次數

## A3寫法

```
當 被點擊  
定位到 x: -120 y: -20  
廣播訊息 說明
```

```
當收到訊息 說明  
詢問 這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。 並等待  
變數 終極密碼 隱藏  
變數 次數 設為 0  
變數 終極密碼 設為 隨機取數 1 到 99  
廣播訊息 開始猜
```

```
當收到訊息 開始猜  
詢問 請問是終極密碼是多少？ 並等待  
變數 次數 改變 1  
如果 詢問的答案 > 終極密碼 那麼  
說出 太大了 持續 2 秒  
如果 詢問的答案 < 終極密碼 那麼  
說出 太小了 持續 2 秒  
如果 詢問的答案 = 終極密碼 那麼  
變數 終極密碼 顯示  
說出 字串組合 你總共猜了 字串組合 次數 次猜中終極密碼。 持續 3 秒
```

# 陽春版04：增加再玩1次

- ▶ 將程式最後的說出，改為詢問，並加上「，按下R鍵再玩1次，其他鍵結束不玩了。」

The image shows a Scratch script for a number guessing game. The script starts with a '當收到訊息' block, followed by an '詢問' block: '這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。' and '並等待'. It then sets variables: '變數 終極密碼' to '隱藏', '變數 次數' to '0', and '變數 終極密碼' to '隨機取數 1 到 99'. A '重複直到' loop contains an '詢問' block: '請問是終極密碼是多少?' and '並等待'. Inside the loop, it checks '如果 詢問的答案 > 終極密碼' and says '太大了 持續 2 秒', or '如果 詢問的答案 < 終極密碼' and says '太小了 持續 2 秒'. It also increments '變數 次數' by 1. After the loop, it shows the '變數 終極密碼'. A second '詢問' block is added: '字串組合 你總共猜了 字串組合 次數 次，猜中終極密碼，按下R鍵再玩1次，其他鍵結束不玩了。' and '並等待'. It then checks '如果 詢問的答案 = R' and broadcasts a message, or '否則' says '再見! 持續 3 秒' and stops.

A close-up of a Scratch '說出' block with the text: '字串組合 你總共猜了 字串組合 次數 次，猜中終極密碼。 持續 3 秒'. The block is highlighted with a red border.

A close-up of a Scratch '詢問' block with the text: '字串組合 你總共猜了 字串組合 次數 次，猜中終極密碼，按下R鍵再玩1次，其他鍵結束不玩了。 並等待'. The block is highlighted with a red border.

# 陽春版04：增加再玩1次

當 被點擊

定位到 x: -78 y: -12

廣播訊息 說明

當收到訊息 說明

詢問 這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。 並等待

變數 次數 設為 0

變數 終極密碼 隱藏

變數 終極密碼 設為 隨機取數 1 到 99

廣播訊息 開始猜

## A4寫法

當收到訊息 開始猜

詢問 請問終極密碼是多少? 並等待

變數 次數 改變 1

如果 詢問的答案 > 終極密碼 那麼

說出 太大了 持續 2 秒

廣播訊息 開始猜

如果 詢問的答案 < 終極密碼 那麼

說出 太小了 持續 2 秒

廣播訊息 開始猜

如果 詢問的答案 = 終極密碼 那麼

變數 終極密碼 顯示

說出 答對了 持續 2 秒

說出 字串組合 你總共猜了 字串組合 次數 次，猜中終極密碼。 持續 3 秒

詢問 再玩1次請按1，其他鍵結束不玩了 並等待

如果 詢問的答案 = 1 那麼

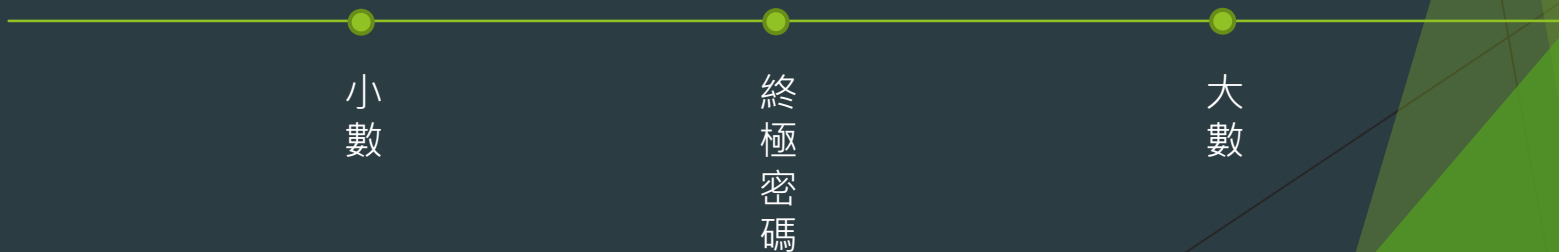
廣播訊息 說明

否則

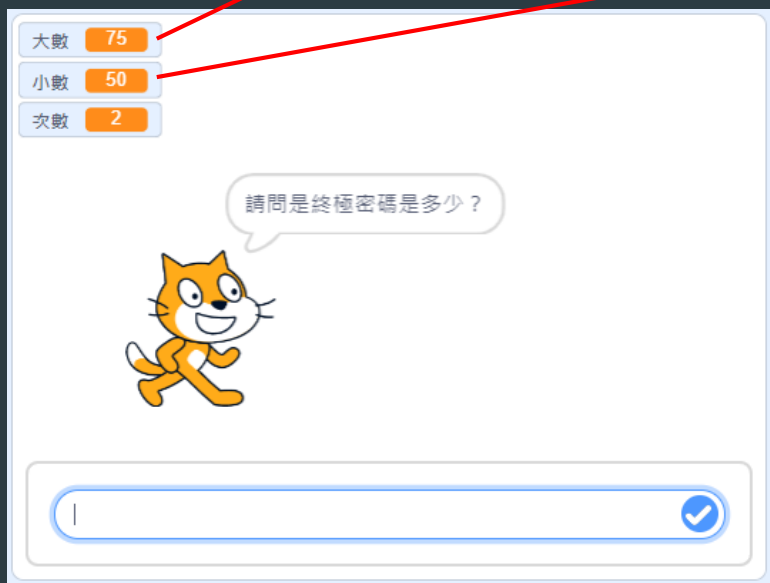
說出 再見 持續 3 秒

# 陽春版05-說明

- ▶ 增加2個變數，紀錄數字範圍  
例如大數、小數
- ▶ 作法：  
如果猜的數字大於終極密碼，將”詢問的答案”設為”大數”，  
如果猜的數字小於終極密碼，將”詢問的答案”設為”小數”。



## 陽春版05-程式碼



當收到訊息 開始猜

詢問 請問是終極密碼是多少？ 並等待

變數 次數 改變 1

如果 詢問的答案 > 終極密碼 那麼

變數 大數 設為 詢問的答案

說出 太大了 持續 2 秒

廣播訊息 開始猜

如果 詢問的答案 < 終極密碼 那麼

變數 小數 設為 詢問的答案

說出 太小了 持續 2 秒

廣播訊息 開始猜

如果 詢問的答案 = 終極密碼 那麼

變數 終極密碼 顯示

詢問 字串組合 你總共猜了 字串組合 次數 次猜中終極密碼，按下R鍵再玩1次，其他鍵結束不玩了。 並等待

如果 詢問的答案 = R 那麼

廣播訊息 說明

否則

說出 再見！ 持續 3 秒

停止 全部

當 被點擊

定位到 x: -120 y: -20

廣播訊息 說明

當收到訊息 說明

詢問 這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。 並等待

變數 終極密碼 隱藏

變數 大數 設為 100

變數 小數 設為 0

變數 次數 設為 0

變數 終極密碼 設為 隨機取數 1 到 99

廣播訊息 開始猜

# 陽春版05- 程式碼

## B5寫法

The image displays a Scratch script for a number guessing game. The script is organized into two main sections. The first section, on the left, handles the initial setup and the first guess. It starts with a '當收到訊息' (When I receive a message) block, followed by an '詢問' (Ask) block: '這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。' (This is a number guessing game, press any key to start the game.) and '並等待' (and wait). Below this, several '變數' (Set variable) blocks are used to initialize the game: '大數' (Big number) is set to 100, '小數' (Small number) is set to 0, '次數' (Attempts) is set to 0, and '終極密碼' (Final password) is set to a random number between 1 and 99. A '重複直到' (Repeat until) loop is used to ensure the user's answer matches the final password. The loop contains an '詢問' (Ask) block: '請問是終極密碼是多少?' (What is the final password?). After the loop, the '次數' (Attempts) variable is incremented by 1. An '如果' (If) block checks if the user's answer is greater than the final password. If true, the '大數' (Big number) variable is set to the user's answer, and a '說出' (Say) block displays '太大了' (Too big) for 2 seconds. Another '如果' (If) block checks if the user's answer is less than the final password. If true, the '小數' (Small number) variable is set to the user's answer, and a '說出' (Say) block displays '太小了' (Too small) for 2 seconds. The second section, on the right, is triggered by a '當被點擊' (When clicked) event. It starts with a '定位到 x: -120 y: -20' (Go to x: -120 y: -20) block, followed by a '廣播訊息' (Broadcast message) block. The main logic begins with a '變數' (Set variable) block for '終極密碼' (Final password) to '顯示' (show). This is followed by an '詢問' (Ask) block: '字串組合“你總共猜了” (String combination "You have guessed a total of")' followed by '次數' (Attempts) and '次猜中終極密碼，按下R鍵再玩1次，其他鍵結束不玩了' (Guess the final password again, press R key to play again, other keys end the game). An '如果' (If) block checks if the user's answer is 'R'. If true, a '廣播訊息' (Broadcast message) block is triggered. Otherwise, a '說出' (Say) block displays '再見!' (Goodbye!) for 3 seconds, followed by a '停止' (Stop) block set to '全部' (all).

```
當收到訊息 說明
詢問 這是1個猜數字的遊戲，按下“任何鍵”遊戲開始。 並等待
變數 終極密碼 隱藏
變數 大數 設為 100
變數 小數 設為 0
變數 次數 設為 0
變數 終極密碼 設為 隨機取數 1 到 99
重複直到 詢問的答案 = 終極密碼
詢問 請問是終極密碼是多少? 並等待
變數 次數 改算 1
如果 詢問的答案 > 終極密碼 那麼
  變數 大數 設為 詢問的答案
  說出 太大了 持續 2 秒
如果 詢問的答案 < 終極密碼 那麼
  變數 小數 設為 詢問的答案
  說出 太小了 持續 2 秒
當被點擊
定位到 x: -120 y: -20
廣播訊息 說明
變數 終極密碼 顯示
詢問 字串組合“你總共猜了” 次數 次猜中終極密碼，按下R鍵再玩1次，其他鍵結束不玩了 並等待
如果 詢問的答案 = R 那麼
  廣播訊息 說明
否則
  說出 再見! 持續 3 秒
  停止 全部
```



# 進階1-如何處理小數的問題1

## 方法1

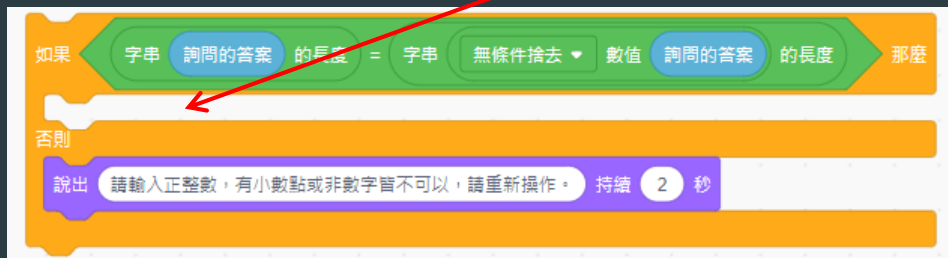
- ▶ 如果輸入的數無條件進位等於輸入的數，繼續執行，否則提醒有小數點，請重新操作，繼續猜？
- ▶ 無法解決例如23.0的.0問題



## 進階2-如何處理小數的問題2

### 方法2

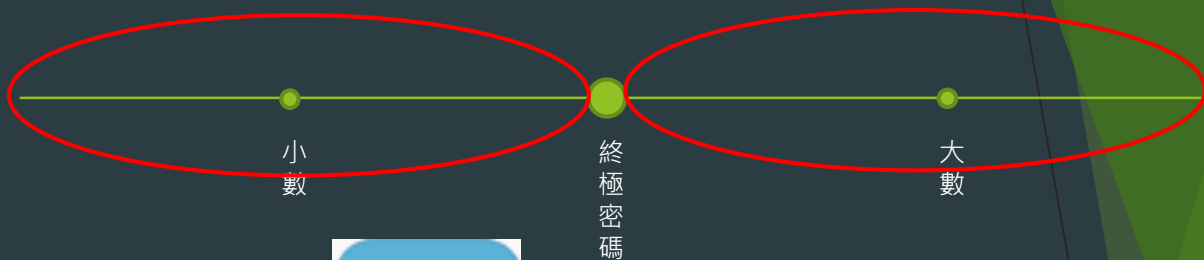
- ▶ 如果輸入的數無條件捨去後的字串長度等於輸入的數的字串長度，繼續執行，否則提醒有小數點，請重新操作，繼續猜？
- ▶ 可以解決有小數點的問題



C1寫法

## 進階2-猜的數字(詢問的答案)沒有在提示的範圍

C2寫法



### 想法1

- ▶ 以下“猜的數字”就是“詢問的答案”
- ▶ 如果“猜的數字”等於“終極密碼”，說出“答對了”，詢問再玩與否？
- ▶ 如果猜的數字小於終極密碼，
  - 1.如果“猜的數字”大於小數，將“猜的數字”設為“小數”，繼續猜。
  - 2.如果“猜的數字”等於小數，說出“猜過這個數了”，繼續猜。
  - 3.如果“猜的數字”小於小數，說出“XXX這個數了”，繼續猜。
- ▶ 如果猜的數字大於終極密碼，
  - 1.如果“猜的數字”小於大數，將“猜的數字”設為“大數”，繼續猜。
  - 2.如果“猜的數字”等於大數，說出“猜過這個數了”，繼續猜。
  - 3.如果“猜的數字”大於大數，說出“XXX這個數了”，繼續猜。

Scratch code for a password guessing game loop. The code starts with a 'Repeat until' block with the condition '詢問的答案 = 終極密碼'. Inside the loop, there is an 'Ask' block: '請問是終極密碼是多少? 並等待'. Below it is a 'Change' block: '次數' (set to 1). Then, an 'If' block: '如果 字串 詢問的答案 的長度 = 字串 無條件請去 數值 詢問的答案 的長度 那麼'. This is followed by two nested 'If' blocks: '如果 詢問的答案 > 終極密碼 那麼' and '如果 詢問的答案 < 終極密碼 那麼'. Below these is an 'Else' block: '否則 說出 請輸入正整數, 有小數點或非數字皆不可以, 請重新操作. 持續 2 秒'. At the bottom of the loop is a 'Show' block: '變數 終極密碼 顯示'.

Scratch code for integer comparison logic. It consists of three 'If' blocks. The first: '如果 詢問的答案 = 大數 那麼 說出 猜過這個數了 持續 2 秒'. The second: '如果 詢問的答案 > 大數 那麼 說出 XXX這個數了 持續 2 秒'. The third: '如果 詢問的答案 < 大數 那麼 變數 大數 設為 詢問的答案 說出 太大了 持續 2 秒'. The third block is circled in red.

擺在最後面

Scratch code for decimal comparison logic. It consists of three 'If' blocks. The first: '如果 詢問的答案 = 小數 那麼 說出 猜過這個數了 持續 2 秒'. The second: '如果 詢問的答案 < 小數 那麼 說出 XXX這個數了 持續 2 秒'. The third: '如果 詢問的答案 > 小數 那麼 變數 小數 設為 詢問的答案 說出 太小了 持續 2 秒'. The third block is circled in red.

擺在最後面

## 想法2



- ▶ 以下“猜的數字”就是“詢問的答案”。
- 詢問的答案
- ▶ 1.如果“猜的數字”等於終極密碼，說出“答對了”，詢問再玩與否？
  - ▶ 2.如果“猜的數字”等於小數，說出“猜過這個數了。”，繼續猜。
  - ▶ 3.如果“猜的數字”等於大數，說出“猜過這個數了。”，繼續猜。
  - ▶ 4.如果“猜的數字”小於小數，說出“XXX.....”，繼續猜。
  - ▶ 5.如果“猜的數字”大於大數，說出“XXX.....”，繼續猜。
  - ▶ 6.如果“猜的數字”大於小數而且小於終極密碼，將“猜的數字”設為“小數”，繼續猜。
  - ▶ 7.如果“猜的數字”小於大數而且大於終極密碼，將“猜的數字”設為“大數”，繼續猜。

頭暈了嗎？

休息一下