

CH. 8

重複的力量：迴圈

HORAZON

應用程式設計

本章目標

1. 理解 迴圈 (Loop) 的概念
2. 學習 `for each number` 積木
3. 實作：累加計算機 ($1+2+\dots+N$)

為什麼需要迴圈？

如果我要算出 1 加到 10 的總和：

```
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
```

手寫還行。

但如果要算出 1 加到 1000 呢？

程式碼會長到爆炸！

迴圈就是讓電腦：「重複做某件事，直到條件滿足為止」。

FOR 迴圈 (計次迴圈)

在 Control (黃色) 抽屜裡：

```
for each number from [ 1 ] to [ 5 ] by [ 1 ]
do
```

(這邊放要重複做的事)

- **from 1:** 開始數字
- **to 5:** 結束數字
- **by 1:** 每次增加多少 (間隔)

這段程式會執行 5 次，每次會有一個變數 **number** 分別是 1, 2, 3, 4, 5。

實作：累加器 (1 加到 N)

畫面

- 輸入 N
- 按鈕「計算」
- 結果標籤

變數

- sum (總和)：一開始一定要設為 0！

程式邏輯

當 按鈕被點選

設 sum 為 0 (歸零很重要！)

for each i from 1 to (輸入的數字) by 1

設 sum 為 (sum + i)

設 標籤.文字 為 sum

追蹤看看 (假設輸入 3)

$$1. \text{ i=1, sum} = 0 + 1 = 1$$

$$2. \text{ i=2, sum} = 1 + 2 = 3$$

$$3. \text{ i=3, sum} = 3 + 3 = 6$$

結束 -> 答案 6。

實作：九九乘法表

如果我們用兩個迴圈套在一起 (巢狀迴圈)...

```
for i from 1 to 9
    for j from 1 to 9
        顯示 (i * j)
```

這樣就可以印出整個九九乘法表了！(這對初學者比較難，我們先玩累加就好)

WHILE 迴圈 (條件迴圈)

如果不確定做幾次，只知道「當某條件成立時繼續做」，就用 `while`。

例如：**一直擲骰子，直到擲出 6點為止**。你不知道要丟幾次，這時候就適合用 While。

(AI2 中 While 積木也在 Control 裡，小心不要寫成無窮迴圈，手機會當機！)

重點回顧

- For Loop: 知道次數時使用 (例如 1~100)。
- 變數歸零: 累加之前，記得把桶子清空。
- Accumulator (累加器): `sum = sum + i` 是最常見的寫法。

下一章：APP 的分身術！多重畫面！