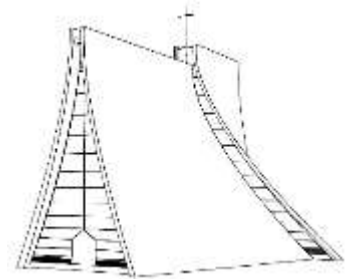




運算思維與程式設計

CH11 迴圈控制

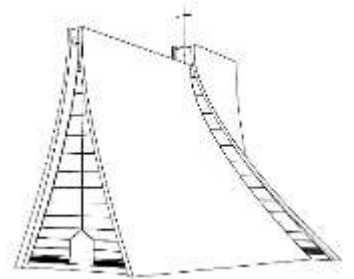


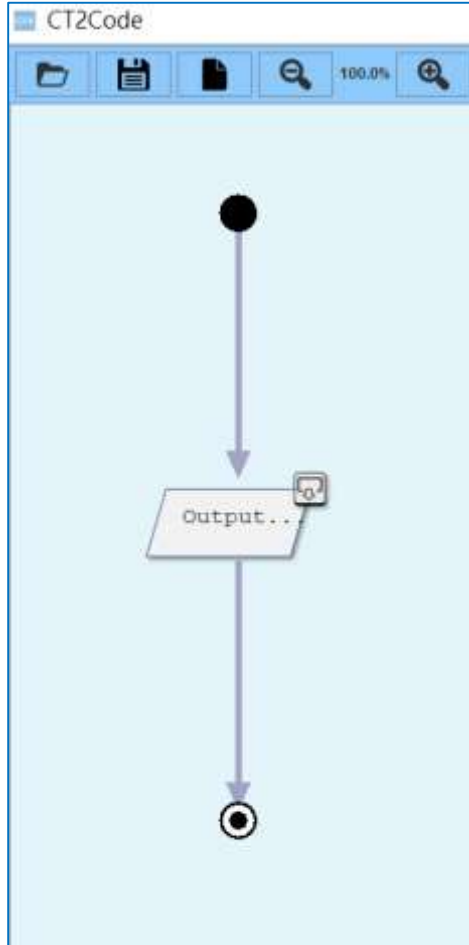


迴圈 (Loop)

迴圈，可是程式最好用的功能之一，可以幫我們省下許多重覆性的工作。

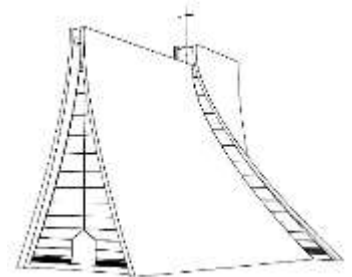
什麼是迴圈呢？每年在地球上的我們，都會經歷一個最大的迴圈，那就是地球繞著太陽公轉一週！又或著地球每24小時都會自轉一圈。

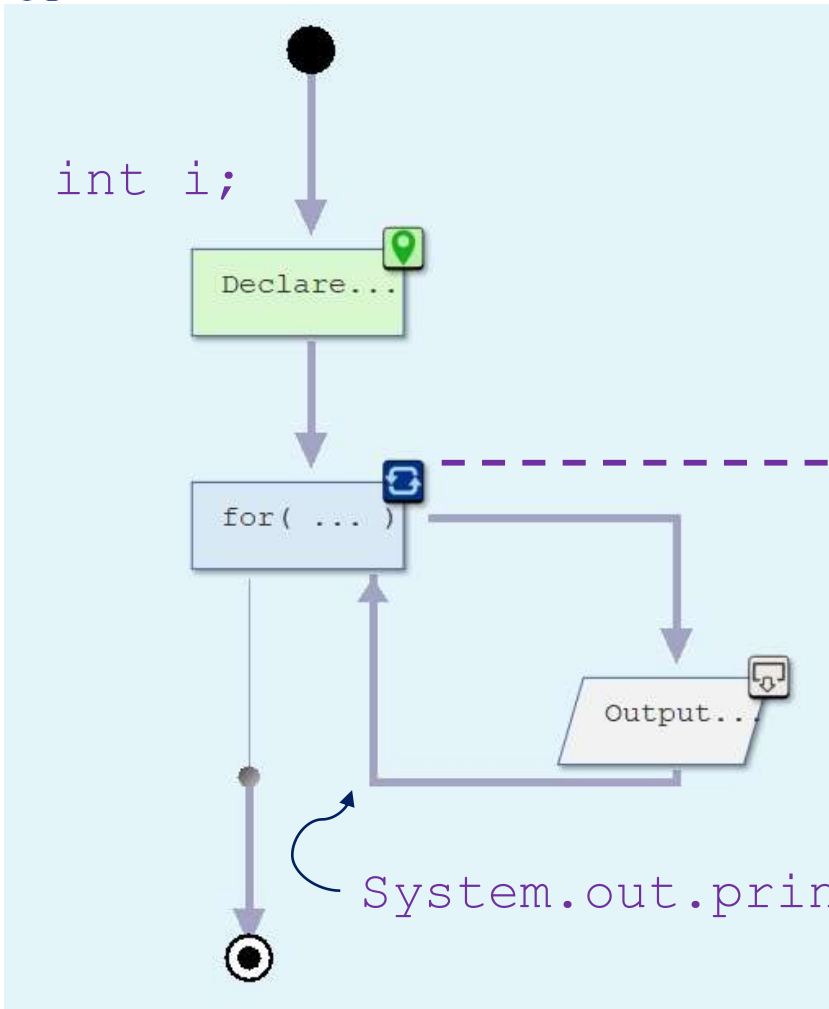




`System.out.print("有夢最美!");`

上述字串輸出一次很容易，
如果要輸出10次，甚至100
次時，難到要重覆寫100次
的輸出指令嗎？



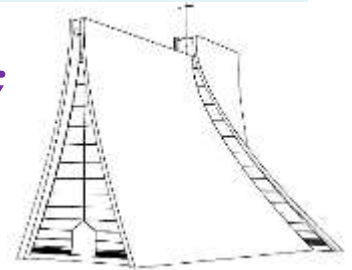


輸出 5 次「有夢最美!」字串!

For迴圈設定



i (選擇變數)	<input type="checkbox"/> 變數是否為陣列
陣列索引	遞增
1 (開始值)	1 (遞增或遞減值)
5 (結束值)	
確定 取消	

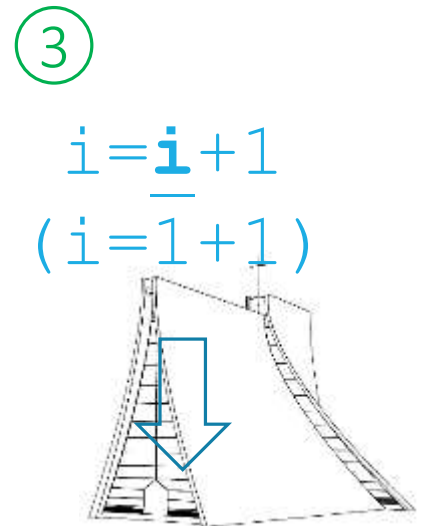
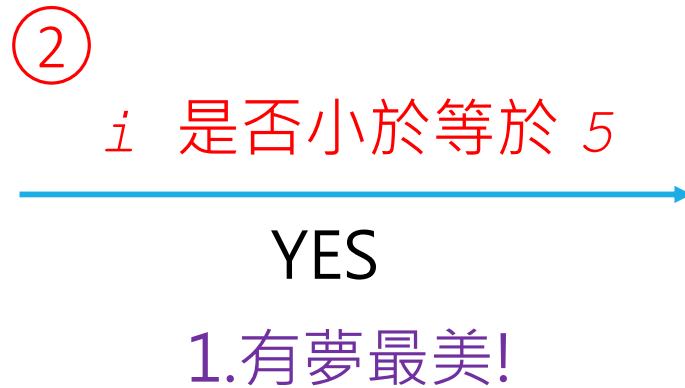
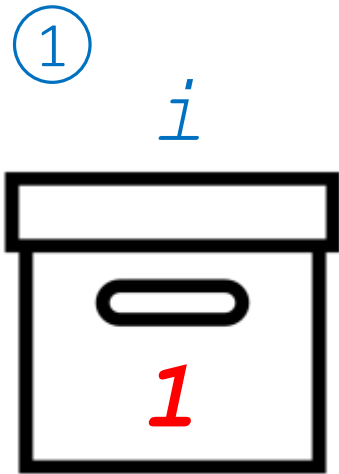




```
int i;  
for ( ① i = 1; ② i <= 5; ③ i = i + 1 )  
{  
    System.out.println(i+"有夢最美!");  
}
```

(起始值) (條件式) (遞增值)

(結束值)

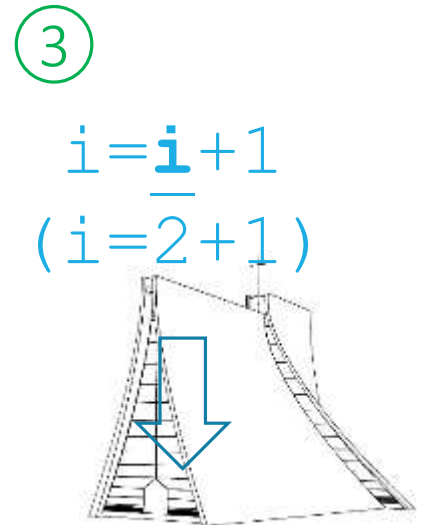
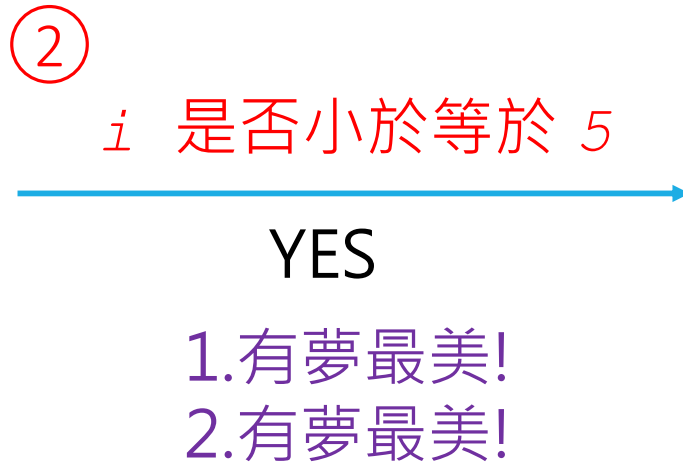
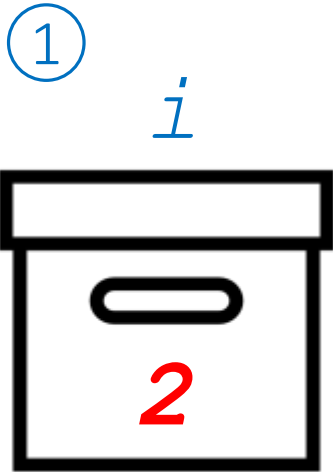




```
int i;  
for ( ① i = 1; ② i <= 5; ③ i = i + 1 )  
{  
    System.out.println(i+"有夢最美!");  
}
```

(起始值) (條件式) (遞增值)

(結束值)





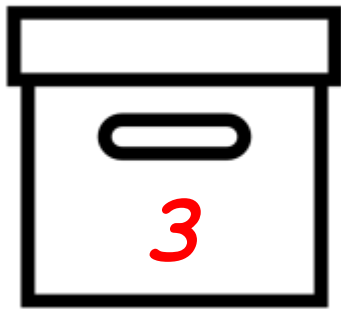
```
int i;  
for ( ① i = 1; ② i <= 5; ③ i = i + 1 )  
{  
    System.out.println(i+"有夢最美!");  
}
```

(起始值) (條件式) (遞增值)

(結束值)

①

i



②

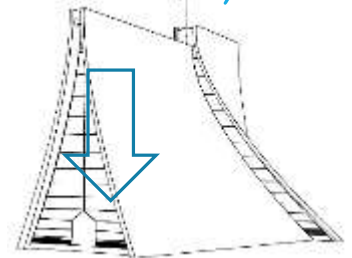
i 是否小於等於 5

YES

- 1.有夢最美!
- 2.有夢最美!
- 3.有夢最美!

③

$i = \underline{i} + 1$
($i = 3 + 1$)





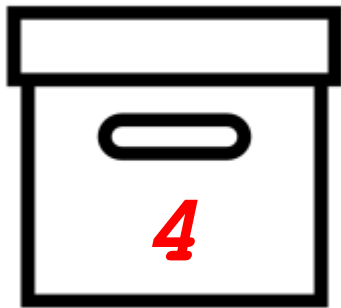
```
int i;  
for ( ① i = 1; ② i <= 5; ③ i = i + 1 )  
{  
    System.out.println(i+"有夢最美!");  
}
```

(起始值) (條件式) (遞增值)

(結束值)

①

i



②

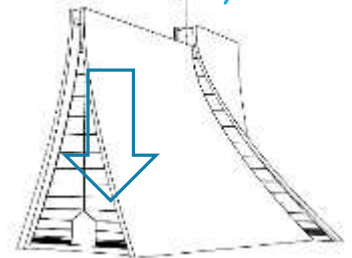
i 是否小於等於 5

YES

- 1.有夢最美!
- 2.有夢最美!
- 3.有夢最美!
- 4.有夢最美!

③

$i = \underline{i} + 1$
($i = 4 + 1$)

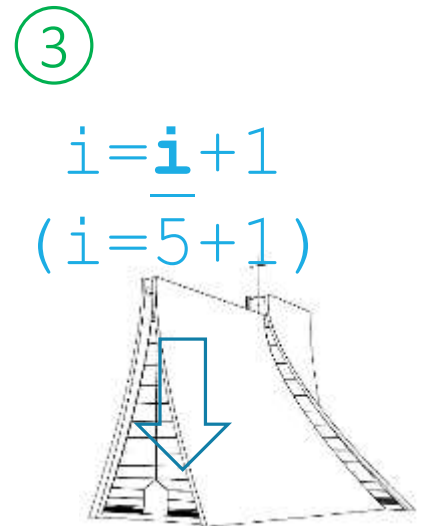
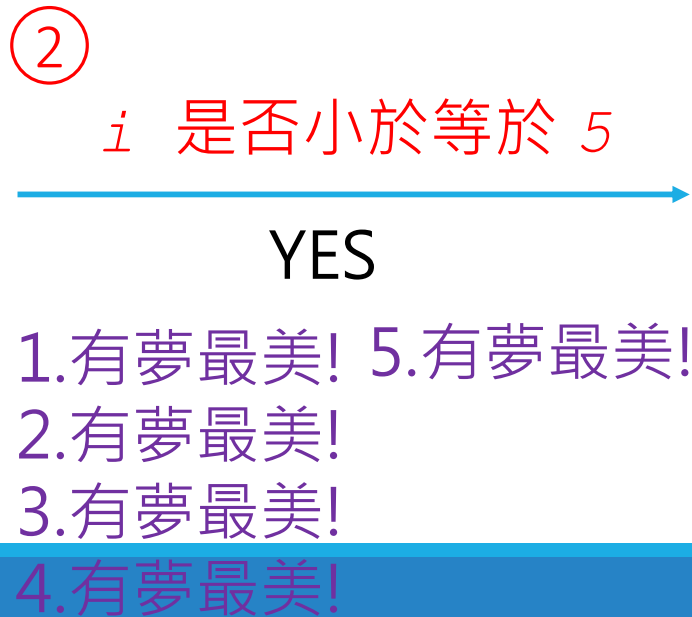
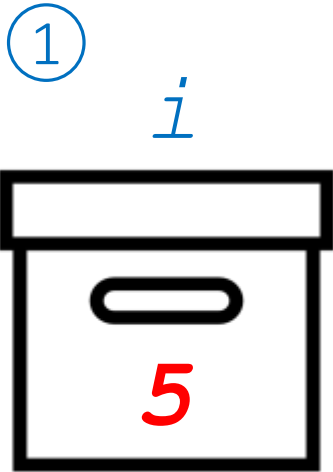




```
int i;  
for ( ① i = 1; ② i <= 5; ③ i = i + 1 )  
{  
    System.out.println(i + ".有夢最美!");  
}
```

(起始值) (條件式) (遞增值)

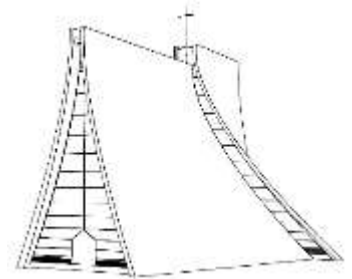
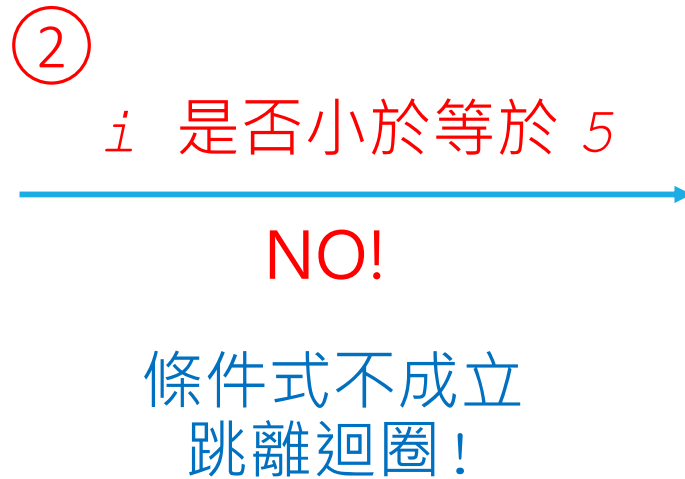
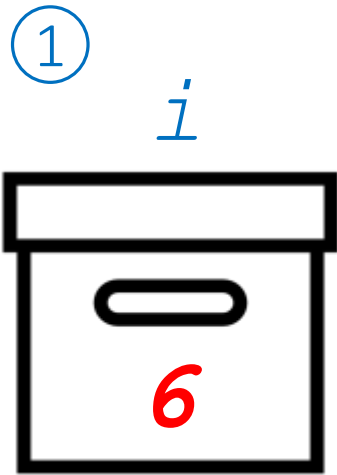
(結束值)





```
int i;  
for ( i = 1; i <= 5; i = i + 1 )  
{  
    System.out.println(i+"有夢最美!");  
}
```

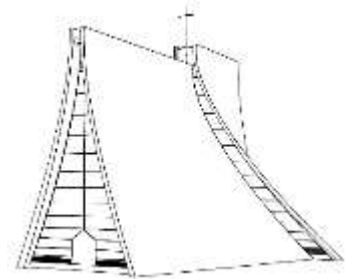
(起始值) (條件式) (遞增值) (結束值)





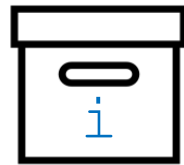
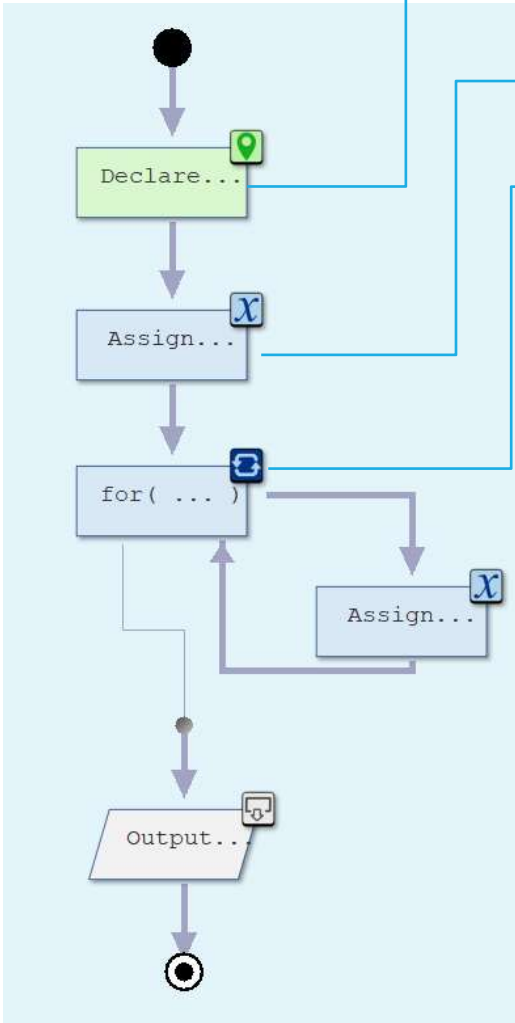
隨堂演練

以迴圈方式，進行 $1+2+3+4+5$ 加總運算，並輸出其最終結果。

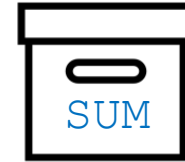




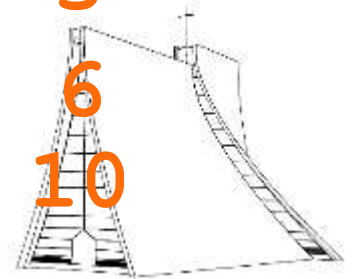
```
int i;  
int sum;  
sum = 0;  
for( i=1; i<=5; i=i+1 )  
{  
    sum = sum + i;  
}  
System.out.println("總合="+sum);
```



1
2
3
4



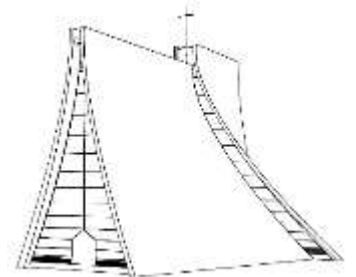
0
3
6
10





隨堂演練

1. 以迴圈方式，進行 $1+2+3+\dots+1000$ 加總運算，並輸出其最終結果。
2. 以迴圈方式，進行 $2+4+6+\dots+2000$ 加總運算，並輸出其最終結果。





```
int i;  
int sum;  
sum = 0;  
for( i=2; i<=2000; i=i+2)  
{  
    sum = sum + i;  
}  
System.out.println("總合="+sum);
```

For迴圈設定

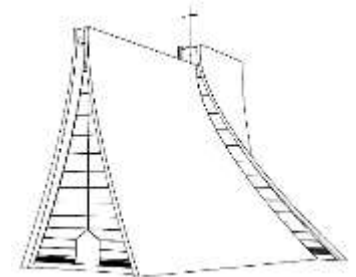
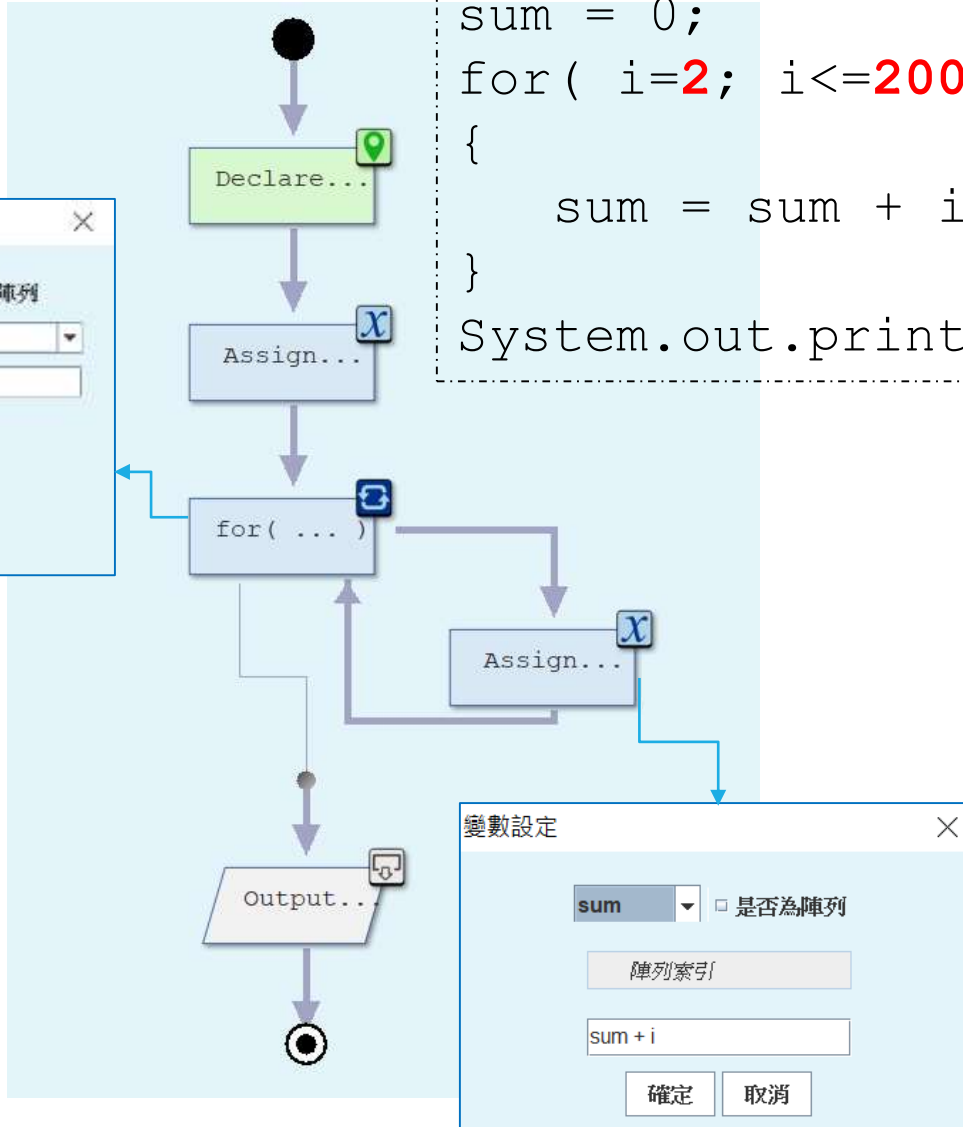
變數是否為陣列

陣列索引: 遞增

0 2

2000

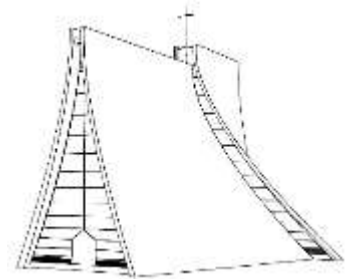
確定 取消





隨堂演練

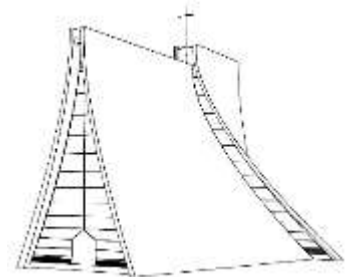
迴圈方式，進行 $2+4+6+\dots+2*N$ 加總運算，其中 N 為使用者任一輸入的一數字，最後輸出總合結果。





隨堂演練

1. 請列出 1-1000 中所有可以被5整除的數?
2. 請列出1~1000中所有可以被2與3同時整除的數?





迴圈結構選擇

事前已知道要進行
多少次的迴圈
(Known number
of iterations)

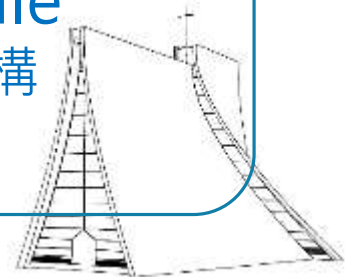


for
結構

事前不知道要進行
多少次的迴圈
(Unknown
number of
iterations)

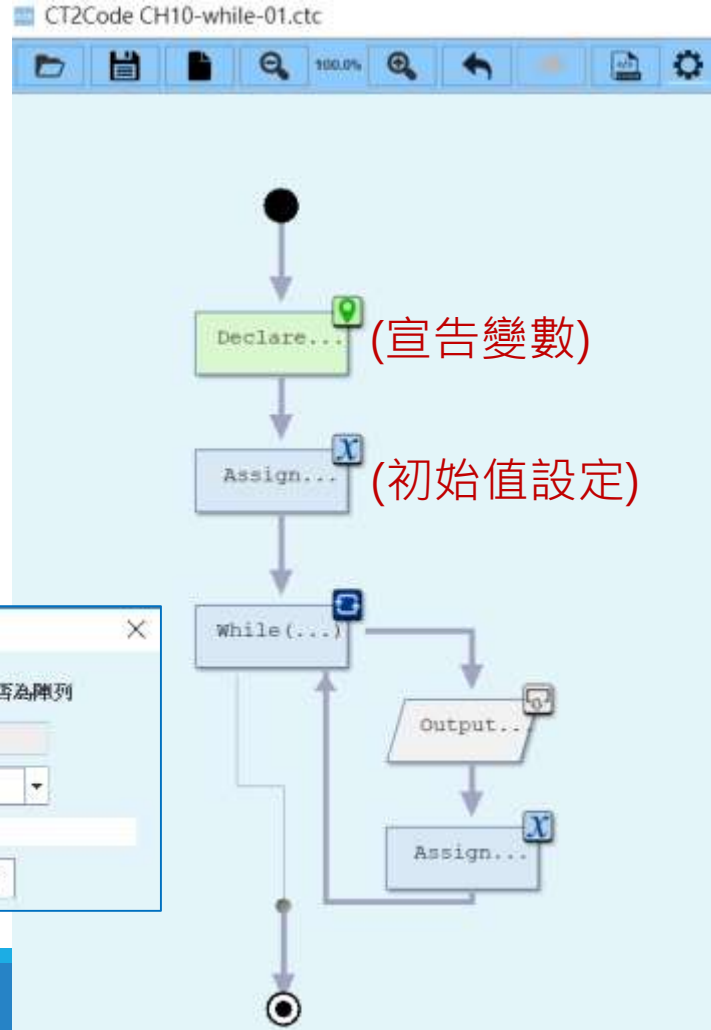


while
結構





While 迴圈介紹



While迴圈設定

變數是否為陣列

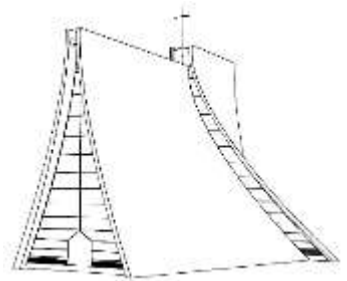
陣列索引

5

確定 取消

條件為真時，執行迴圈內的程式。

```
int i;
i = 0;
while( i<=5 )
{
    System.out.println("TEST!");
    i = i+1;
}
```

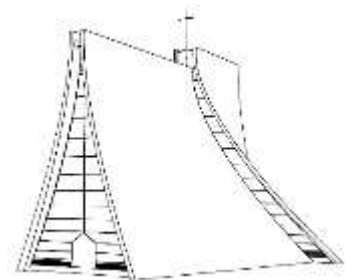
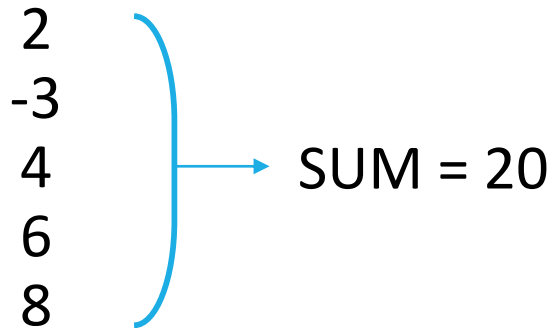




隨堂演練

1. 假設有一條繩子長13500公尺，每天剪去一半的長度，請問需要花費幾天後的时间，繩子的長度才會小於5公尺？
2. 撰寫一支程式，提示使用者輸入5個數字，當輸入第5個數字後，程式會自動加總所有正數，並輸出總合？

例如：





隨堂演練

3. 輸入一數字N，判斷該數是否為質數？

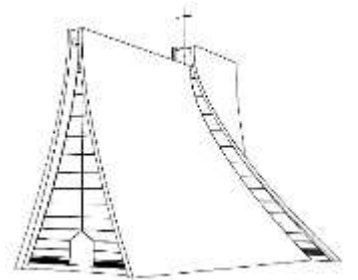
例如：

請輸入一數字：

你所輸入的數字為 13，它是質數。

請輸入一數字：

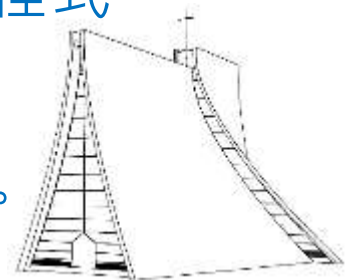
你所輸入的數字為 100，它不是質數





隨堂演練

1. 有 7 位朋友到知名美式餐廳聚餐慶生，此餐廳採點餐的計算方式。
2. 程式執行時，如執行結果參考畫面，畫面顯示【請輸入套餐價錢（-1 離開系統）】，請使用者輸入第一道餐點費用，再分別依序要求輸入其他道餐點的費用。
3. 若要停止執行程式計算，需輸入-1後停止程式計算，-1不列入餐點計算費用內。
4. 請輸出餐點數量、總費用及平均負擔費用。



Thank You!