

Chapter7-迴圈

for 迴圈

語法:

1

2

4

```
for( 設定迴圈初始值 ; 判斷條件 ; 設定增減量){
```

```
    /* do something */
```

3

```
}
```

```
int i, sum = 0;
```

```
for( i = 1 ; i <= 10 ; i++){
```

```
    sum += i;
```

```
}
```

- 如果迴圈大括號只有一個 statement 大括號可以省略
- 執行流程:
 1. 先執行迴圈初始值的部份
 2. 執行判斷條件, 若為 true 則執行, 為 false 則停止並跳出迴圈
 3. 執行大括號內的 statement
 4. 執行設定增減量內容之後跳到第 2 步

for 迴圈

```
int i, sum = 0;
```

```
for(i = 1; i <= 10 ; i++){
```

```
    sum += i;
```

```
}
```

```
int i, sum;
```

```
for(i = 1, sum = 0; i <= 10 ; i++){
```

```
    sum += i;
```

```
}
```

```
for(int i = 1, sum = 0; i <= 10 ; i++){
```

```
    sum += i;
```

```
}
```

- 可以在設定迴圈初始值的部份宣告並初始化該變數, 也可以進行其他運算(例如賦值)
- 但要注意 scope 的問題

while 迴圈

語法:

1

```
while ( 判斷條件){
```

```
    /* do something */
```

2

```
}
```

```
int i = 1, sum = 0;
```

```
while( i <= 10){
```

```
    sum += i;
```

```
    i++;
```

```
}
```

- 如果迴圈大括號只有一個 statement 大括號可以省略
- 執行流程:
 1. 執行判斷條件, 若為 true 則執行, 為 false 則停止並跳出迴圈
 2. 執行大括號內的 statement 跳到第 1 步

無窮迴圈 endless loop

```
int i = 0;
while ( i >= 0 ){
    printf("%d", i++);
}
```

- 無窮迴圈為迴圈永遠無法結束，並一直執行
- 無窮迴圈與**迴圈控制變數**以及**控制變數增減量的運算**有關
- 無窮迴圈對於演算法而言是不好的情況，但對於某些系統以及應用來講卻非常合適 e.g. 遊戲，作業系統

do while 迴圈

```
int i = 1, sum = 0;
```

```
do {
```

```
    sum += i;
```

1

```
    i++
```

```
} while( i <= 10);
```

2

- 如果迴圈大括號只有一個 statement 大括號可以省略
- 執行流程:
 1. 執行大括號內的 statement
 2. 執行判斷條件, 若為 true 則執行跳到第 1 步, 為 false 則停止並跳出迴圈
- do while 與 while 迴圈最大差別是, do while 事先做再檢查, while 是先檢查再做
- do while 內部執行次數 ≥ 1 , 而 while 則是 ≥ 0

空迴圈

```
for(int i = 0 ; i <= 10 ; i++)  
    ;
```

```
int i = 0;
```

```
while(i <= 10);  
    i++;
```

- 程式執行空迴圈不會做任何事情，但卻會佔用CPU的資源
- 初學者容易在迴圈外無意間多加個『;』導致迴圈變成空迴圈
- 在 while 以及 do while 迴圈的空迴圈容易形成無限迴圈

巢狀迴圈

- 定義: 迴圈裡出現另一個迴圈

九九乘法表:

```
for(int i = 1 ; i <= 9 ; i++){    /* 外部迴圈 */
    for(int j = 1 ; j <= 9 ; j++){    /* 內部迴圈 */
        printf("%d %d=%2d ", i, j, i*j);
    }
    printf("\n");
}
```


迴圈的跳離

- 使用 break, continue, goto(不建議) 敘述來進行跳出、執行下一迭代...等等迴圈控制
- 常搭配在條件控制 if 後方來進行各種操作

迴圈的跳離 - break

```
int i = 0;
for(int i = 0 ; i <= 10 ; i++){
    if( i == 3) break;
    printf("%d ", i);
}
```

- 當執行 break 敘述時, 迴圈會結束跳出
- 因此左邊的例子來看只會列印出 0 1 2
並不會列印出 3 4 5 6 7 8 9 10

迴圈的跳離 - continue

```
// 只列印出 0 ~ 100 中的奇數部份
for(int i = 0 ; i <= 100 ; i++){
    if( i % 2 == 0) continue;
    printf("%d ", i);
}
```

- 當執行 continue 敘述時, 迴圈會結束該次執行步驟並且執行下一次迭代
- continue: 1, 3, 5, 7, 9
- break: (無)

課堂練習題

- zerojudge - a004. 文文的求婚
- zerojudge - a038. 數字翻轉
- zerojudge - c013. 00488 - Triangle Wave