流程控制

邏輯分析與程式設計

國立雲林科技大學
王照明老師

助教： 徐偉智、李柏廷、李亭儀

流程控制的基礎

基礎說明

- 一般來說，JavaScript程式碼大部分都是一列程式敘述接著一列程式敘述循序的執行，但是對於複雜工作，為了達成預期的執行結果，我們需要使用「流程控制結構」（Control Structures）來改變執行順序。

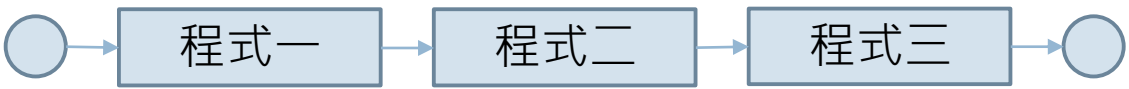
基礎說明

- 不同的程式語言所提供的流程控制指令也會隨之不同，但一般可以分為以下三種：

1. 繼續執行位在不同位置的一段指令（無條件分支指令）。
2. 若特定條件成立時，執行一段指令，例如C語言的switch指令，是一種有條件分支指令。
3. 執行一段指令若干次，直到特定條件成立為止，例如C語言的for指令，仍然可視為一種有條件分支指令。

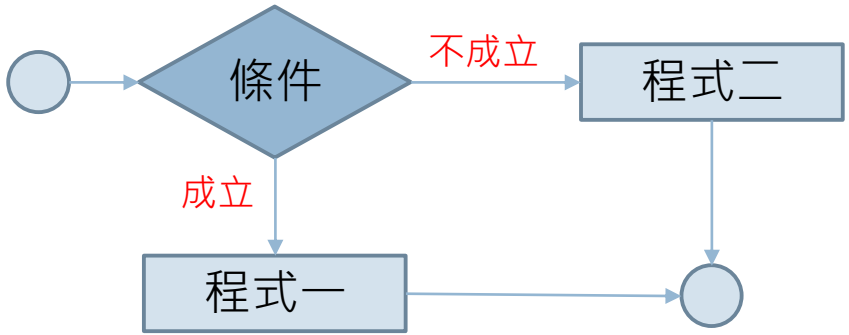
循序結構（Sequential）

- 循序結構是程式預設的執行方式，其執行步驟是一步一步承續下來，且依先後順序表示，第一個程式執行完才會換下一個，如下圖所示：



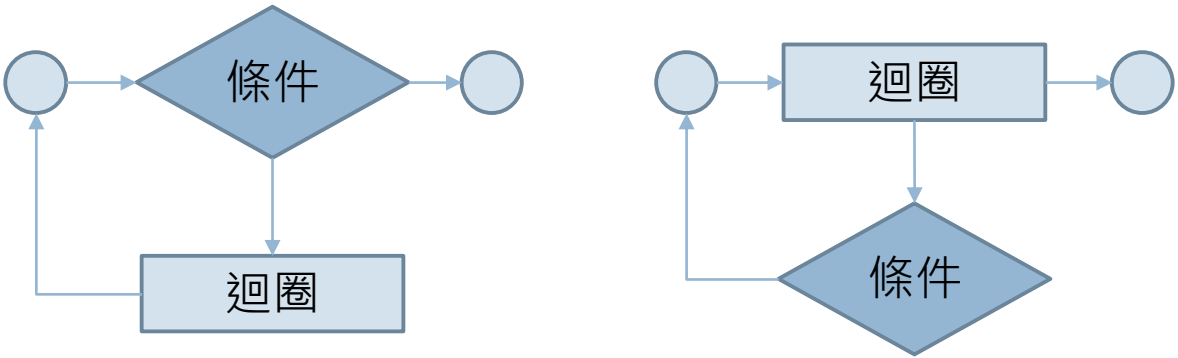
選擇結構（Selection）

- 選擇結構是一種條件判斷，程式流程進入判斷區域後，會判斷測試條件是否成立。然後，依據判斷的結果選擇程式的流向，若是成立則執行程式一，若不成立則跳到程式二。如下圖所示:



重複結構（Iteration）

- 重複結構是迴圈控制，可以重複執行一個程式區塊的程式碼，根據結束條件結束迴圈，依結束條件測試的位置不同分為兩種：前測式重複結構是在進入迴圈前測試條件（左圖），和後測式重複結構是在迴圈結束測試條件（右圖）。



邏輯與比較運算子

比較運算子

- JavaScript的比較運算子可以比較2個運算元是相等、不相等、哪一個比較大或哪一個比較小，其說明和範例如下表所示：

運算子	說明	運算式範例	運算結果
==	等於	4 == 3	false
!=	不等於	4!= 3	true
<	小於	4 < 3	false
>	大於	4 > 3	true
<=	小於等於	4<= 3	false
>=	大於等於	4 >=3	true

邏輯運算子

- 如果條件不只一個，而是多個比較運算式，我們需要使用邏輯運算子來連接多個比較運算式，其說明如下表所示：

運算子	說明
!	NOT，傳回運算元相反的值，true成false；false成true
&&(和)	if(i>=5 && i<=10)即表示判斷變數i是否在5~10之間
||(或)	if(i==5 || i==10)即表示判斷變數i是否等於5或10

條件判斷

if單選條件敘述

- JavaScript的if條件敘述是一種是否執行的單選題，可以決定是否執行程式區塊內的程式碼，如果條件運算結果為true，就執行括號之間的程式碼。
- 當條件式為真時，執行大括號內的動作，如果要執行的動作，只有一項，大括號可以省略。

```
if (num<0){
   document.write( “輸入的值為負數<br/>");
}
```

- 若條件式結果為ture，執行程式敘述區塊
- 若條件式結果為false，則跳過不執行敘述區塊

if/else二選一條件敘述

- 如果擁有兩個程式區塊，而且只能二選一，執行其中一個程式區塊，若條件式為true時，執行區塊敘述一，否則(條件式為fasle時)，執行區塊敘述二。例如：判斷成績是否及格，就是只能二選一

```
if (Grade >= 60) {
  document.write(“恭喜及格=” + Grade + “<br/>”);//區塊一
}
else {
  document.write(“不及格=” + Grade + “<br/>”);//區塊二
}
```

if/else多選一條件敘述

- 在JavaScript程式如果需要多選一條件敘述，可以巢狀if/else敘述，例如：公車票是18歲以下購買半票;19~64歲購買全票；65歲以上購買敬老票。

```
if (Age <= 18)
    document.write("購買半票!<br/>");
else if (Age >= 65)
    document.write("購買敬老票!<br/>");
else
    document.write("購買全票!<br/>");
```

- 當條件式一結果為true時，則執行區塊敘述一，後面就不繼續進行判斷。
- 利用此結構能避免重複判斷，改善效能問題

if/else多選一條件敘述

- 以下程式結果相同，皆能判別出目前所輸入數字的正負，但效能不同。

```
if(num==0){
                document.write( “此數字為0");
}if(num<0){
                document.write( “此數字為負數");
}if(num>0) {
                document.write( “此數字為正數");
}
```

上圖使用多個判斷式，下圖使用使用If/else多層巢狀結構

```
if(num==0){
                document.write( “此數字為0");
}else if(num<0){
                document.write( “此數字為負數");
}else {
                document.write( “此數字為正數");
}
```

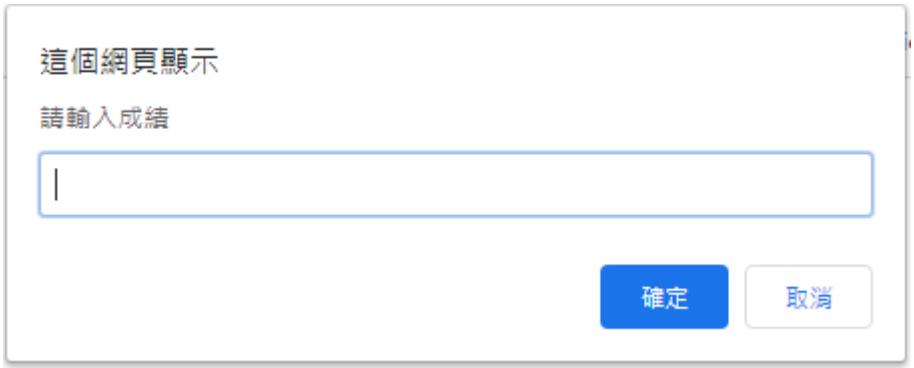
switch多選一條件敘述

- switch多選一條件敘述，直接依照符合條件來執行不同程式區塊的程式碼。
- 鍵值與條件值1比對，如果兩值相等，則執行區段1敘述，並透過break指令，跳離switch敘述
- 若鍵值與條件值1比對，兩值不相等時，則繼續執行區段2敘述，以此類推
- 若鍵值與所有條件直都不相等時，則執行default下的default區段敘述。

```
switch (train){   //鍵值
   case 1:
      document.write( “普悠瑪號!<br/>");
      break;
   case 2:
      document.write( “自強號!<br/>");
      break;
   case 3:
      document.write( “莒光號!<br/>");
      break;
   default:
      document.write( “區間車!<br/>");
}
```

Lab01-判斷成績

- 利用if/else來進行成績判斷



Lab01-判斷成績

```
<script>
 var res = window.prompt("請輸入成績");
 if(res==''){
     document.write("此區域不可為空白");
 }
 else if(res<60){
     document.write("不及格");
 }
 else if(res>=60 && res<70){
     document.write("D");
 }
 else if(res>=70 && res<80){
   document.write("C");
 }
```

Lab01-判斷成績

```
  else if(res>=70 && res<80){
    document.write("C");
  }
  else if(res>80 && res<=90){
      document.write("B");
  }
  else if(res>90 && res<=100){
      document.write("A");
  }
  else{
      document.write("請輸入100以內的數字");
  }
</script>
```

迴圈控制

for迴圈敘述

- for迴圈稱為計數迴圈，可以用來執行一段程式碼，當變數值與設定的條件符合時，就會持續執行到條件不符合才停止執行。

```
for(起始式;條件式;步進式){//區塊內敘述}
```

- 起始式
  進入迴圈時，一開始執行的程式運算式，只執行一次。起始式為設定控制迴圈執行變數的起始值。
- 條件式
  判斷是否執行迴圈內區塊的依據。
  如果條件式結果為ture，則繼續執行「區塊內的敘述」。如果條件式結果為false，則跳離迴圈。
- 步進式
  每經過一次迴圈，就會執行一次的運算式
  步進式設定控制迴圈執行變數的遞增或遞減的運算式

for迴圈敘述

```
for(起始式;條件式;步進式){//區塊內敘述}
```

```
for (i = 1; i <= 5; i++) {
   document.write("整數: " + i + "<br/>");
   intSum += i;
}
   document.write("總和: " + intSum + "<br/>");
```

整數: 1
整數: 2
整數: 3
整數: 4
整數: 5
總和: 15

while迴圈敘述

- while迴圈敘述需要自行在程式區塊內處理計數器的增減，迴圈是在開頭檢查結束條件，如果條件true才能夠進入迴圈；false離開迴圈
- 與for迴圈不同的地方
  起始式可放在程式之前
  步進式則放在區塊敘述最後面

```
//起始式
while(條件式){
                    //區塊內敘述
                    //步進式
}
```

```
j=0;
while(j <= 4) {
   document.write("整數: " + j + "<br/>");
   j++;
}
```

整數: 0
整數: 1
整數: 2
整數: 3
整數: 4

do/while迴圈敘述

- 常用的 JavaScript while 迴圈是先判斷條件才執行，而 do while 則是先執行後判斷條件，可以用來執行重覆性的工作項目。

```
do{
    //區塊內敘述
    //步進式
}while(條件式);
```

```
i=0;
   do{
      document.write("整數: " + i + "<br/>");
      i++;
   }while(i<6);
```

整數: 0
整數: 1
整數: 2
整數: 3
整數: 4
整數: 5

跳出迴圈-break

- 當某些條件成立時，可以使用break關鍵字強迫終止迴圈的執行，如同switch條件使用break關鍵字跳出程式區塊
- 範例：當列印至3時則跳出迴圈

```
i=0;
   while (i < 6) {
   if (i == 3)
   break;
   i++;
   }
   document.write("整數: " + i + "<br/>");
```

整數: 3

繼續迴圈-continue

- continue關鍵字可以馬上繼續下一次迴圈的執行，不過，它並不會執行程式區塊中位在continue關鍵字之後的程式碼
- 範例:i==3時，繼續下一次迴圈執行，所以3沒有被印出。

```
i=0;
   for (i = 0; i < 5; i++) {
   if (i == 3) {
      continue;
   }
   document.write("整數: " + i + "<br/>");
   }
```

整數: 0
整數: 1
整數: 2
整數: 4

巢狀迴圈

- 巢狀迴圈是指在迴圈之中擁有其他迴圈，例如：在for迴圈之中擁有for、while或do/while迴圈。
- 通常巢狀的for迴圈可運用在二維平面的運用

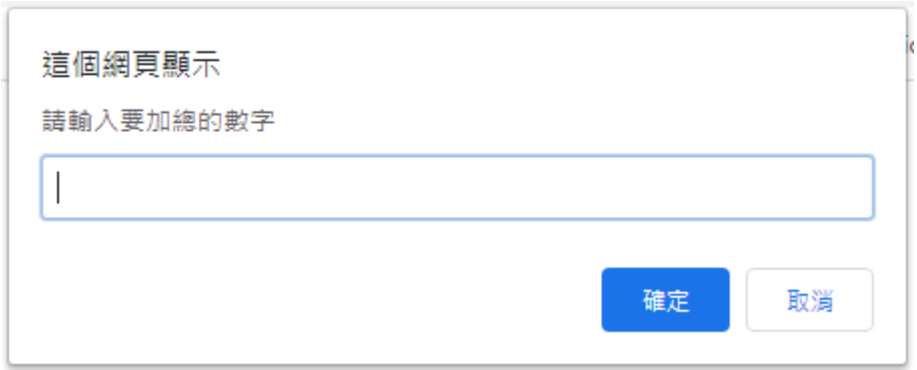
```
var i,j,r3="";
   for(i=1;i<10;i++){
   j=1;
   while(j<10){
      r3 += i+"*"+j+"="+i*j+" ";
      j++;
   }
   r3+="<br/>";
   }
   document.write(r3);
```

巢狀迴圈

1*1=1 1*2=2 1*3=3 1*4=4 1*5=5 1*6=6 1*7=7 1*8=8 1*9=9
2*1=2 2*2=4 2*3=6 2*4=8 2*5=10 2*6=12 2*7=14 2*8=16 2*9=18
3*1=3 3*2=6 3*3=9 3*4=12 3*5=15 3*6=18 3*7=21 3*8=24 3*9=27
4*1=4 4*2=8 4*3=12 4*4=16 4*5=20 4*6=24 4*7=28 4*8=32 4*9=36
5*1=5 5*2=10 5*3=15 5*4=20 5*5=25 5*6=30 5*7=35 5*8=40 5*9=45
6*1=6 6*2=12 6*3=18 6*4=24 6*5=30 6*6=36 6*7=42 6*8=48 6*9=54
7*1=7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49 7*8=56 7*9=63
8*1=8 8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64 8*9=72
9*1=9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81

Lab02-數字加總

- 利用for迴圈來進行加總

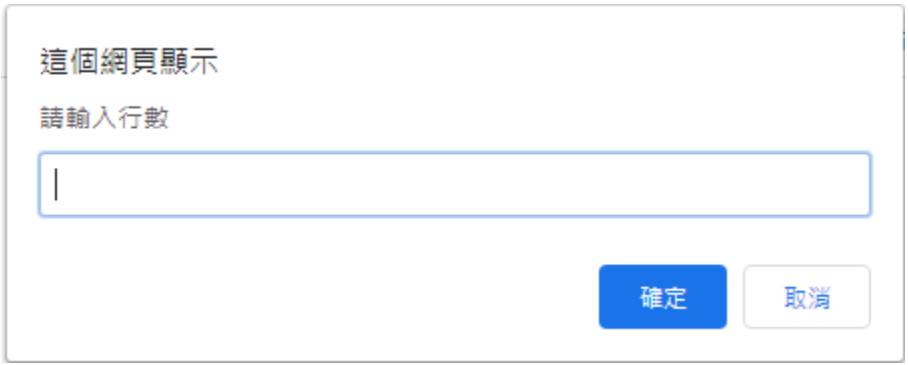


Lab02-數字加總

```
<script>
    var num = window.prompt("請輸入要加總的數字");
    var sum=0;
    for (i = 1; i <= num; i++) {
    sum += i;
    }
    document.write(sum);
</script>
```

Lab03-直角三角形

- 利用巢狀for迴圈來製作直角三角形



Lab03-直角三角形

```
<script>
    var num = window.prompt("請輸入行數");
    for(var i=0;i<num;i++){ //有幾行
        for(var j=0;j<=i;j++){ //第幾行就印幾個
            document.writeln("*");
        }
        document.writeln("<br/>");//完成一行就換行
    }
</script>
```