



Chapter 03

JavaScript流程控制

中原大學 資訊管理學系

賴錦慧 老師

chlai@cycu.edu.tw

大綱

- 流程控制的基礎
- JavaScript的條件控制
- JavaScript的迴圈控制
- 繼續和跳出迴圈
- JavaScript的巢狀迴圈

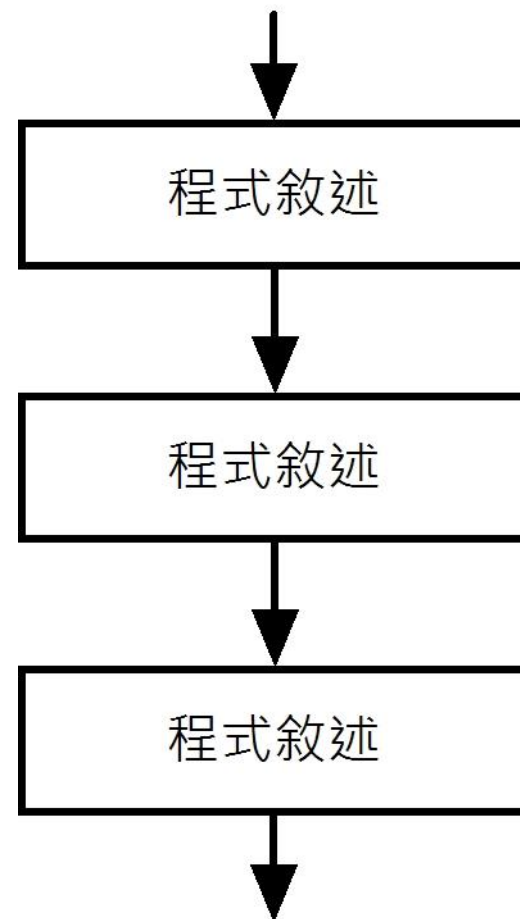
流程控制的基礎

- JavaScript程式碼大部分是循序執行
- 對於複雜的工作，為了達成預期的執行結果，需要使用「流程控制結構」(Control Structures) 來改變執行的順序。

流程控制的基礎-循序結構

循序結構 (Sequential)

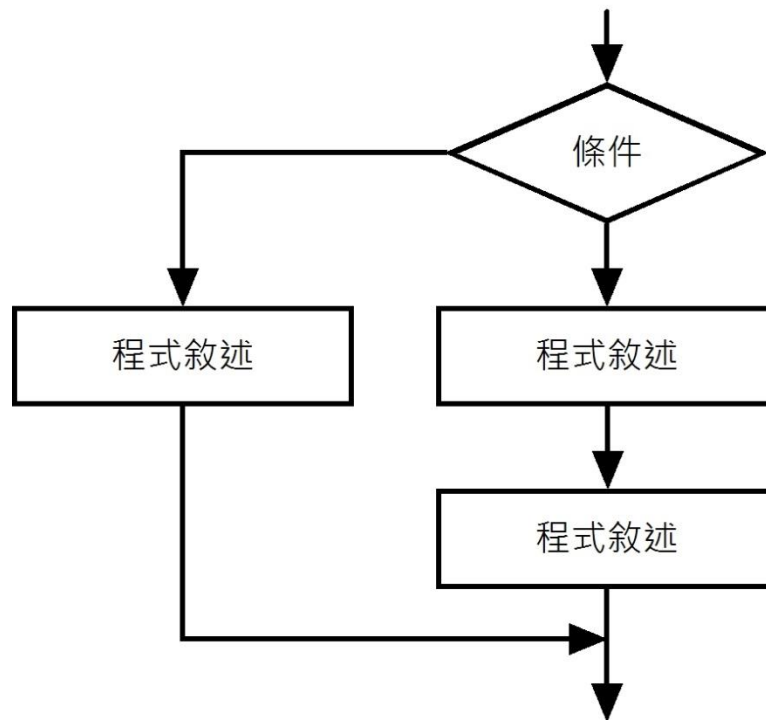
□ 程式預設的執行方式



流程控制的基礎-選擇結構

選擇結構 (Selection)

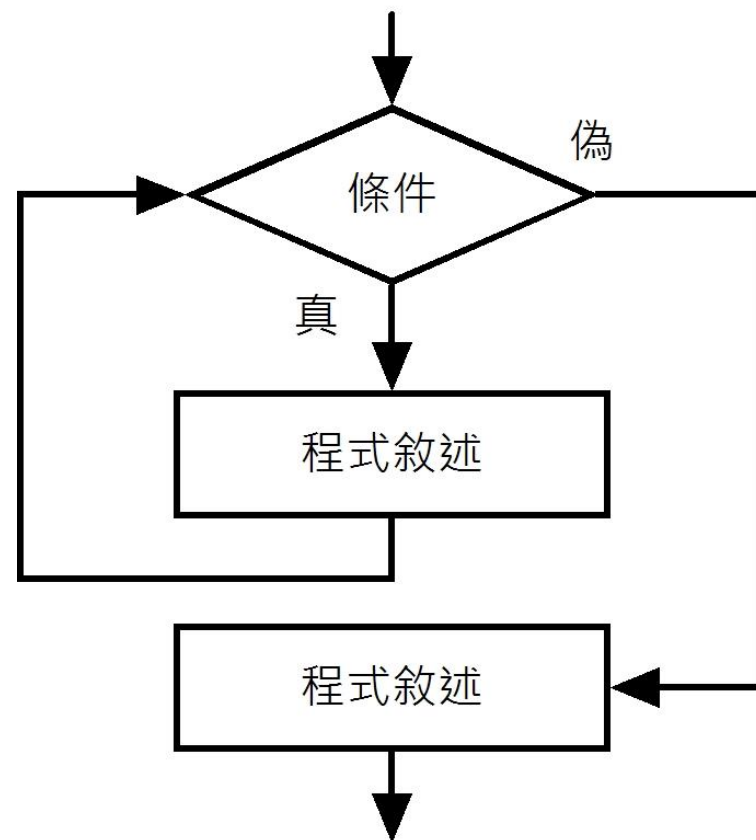
- 一種條件控制
- 依照邏輯或比較運算式的條件，決定執行哪一個程式區塊的程式碼



流程控制的基礎-重複結構1

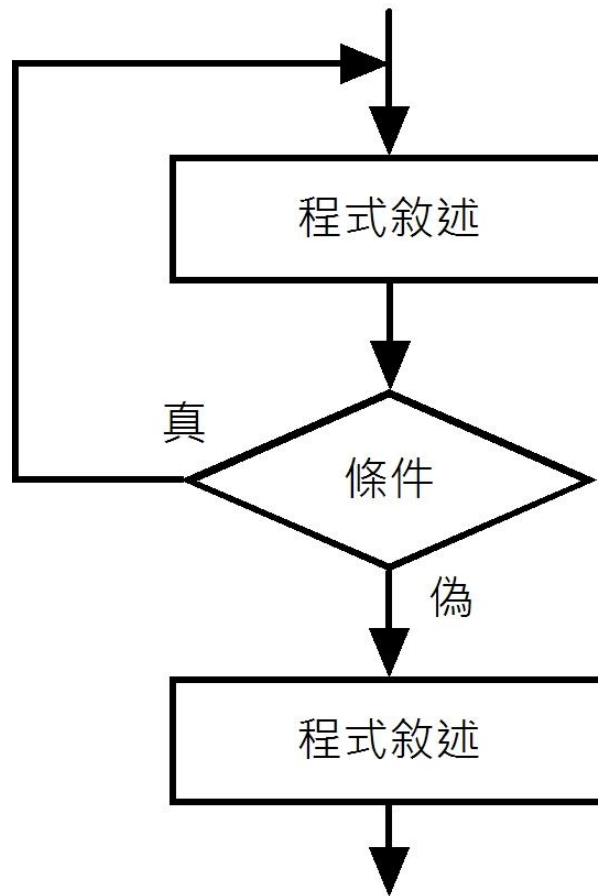
重複結構 (Iteration)

- 迴圈控制
- 可以重複執行一個程式區塊的程式碼，提供結束條件結束迴圈的執行
- 前測式重複結構



流程控制的基礎-重複結構2

- 後測式重複結構
 - 迴圈的程式區塊至少會執行一次



JavaScript的條件控制



- if 是否選擇條件敘述
- if/else 二選一條件敘述
- if/else 多選一條件敘述
- switch 多選一條件敘述
- 條件運算子?:

if--選擇條件敘述

- if條件敘述是一種是否執行的單選題

```
if (strGender == "男"){  
    document.write("男性網友您好! ");  
    document.write("歡迎使用JavaScript<br/>");  
}
```

- 當程式區塊的程式碼只有一列時，可以省略前後的大括號「{」和「}」。

if/else -- 二選一條件敘述

- 如果if條件為true，就執行與else之間的程式區塊，如果為false，就執行else之後的程式區塊

```
if (strGender == "男"){  
    document.write("男性網友您好! ");  
}  
else{  
    document.write("女性網友您好! ");  
}
```

if/else -- 多選一條件敘述

- 重複使用if/else條件敘述來建立多選一條件敘述

```
if (strPayment == "cash")
    document.write("使用現金付款!<br/>");
else
    if (strPayment == "visa")
        document.write("使用VISA信用卡付款!<br/>");
    else
        if (strPayment == "master")
            document.write("使用Master信用卡付款!<br/>");
        else
            document.write("未明的付款方式!<br/>");
```

Switch -- 多選一條件敘述

□ Switch: 依照符合條件來執行不同程式區塊的程式碼

```
switch (strPayment){  
  case "cash":  
    document.write("使用現金付款!<br/>");  
    break;  
  case "visa":  
    document.write("使用VISA信用卡付款!<br/>");  
    break;  
  case "master":  
    document.write("使用Master信用卡付款!<br/>");  
    break;  
  default:  
    document.write("未明的付款方式!<br/>");  
}
```

條件運算子 ?:

- 如同一個if/else條件，使用「？」符號代替if；
「:」符號代替else

```
strHours = (dtHour >= 12) ? " PM" : " AM";
```

JavaScript的迴圈控制



- for迴圈敘述
- for/In迴圈敘述
- while迴圈敘述
- do/while迴圈敘述

for迴圈敘述

- 執行固定次數的程式區塊
- 計數器每次增加或減少一個值，直到迴圈結束條件成立為止

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

for/in迴圈敘述

□ for/in迴圈主要是在顯示物件的所有屬性

```
for (prop in objAddress) {  
    document.write("屬性: " + prop + "=" +  
        objAddress[prop] + "<br/>");  
}
```


練習1

□ 利用迴圈來印出 size 由 1 到 5 的"Hello World!"

```
Font size= 1 ==> Hello World!  
Font size= 2 ==> Hello World!  
Font size= 3 ==> Hello World!  
Font size= 4 ==> Hello World!  
Font size= 5 ==> Hello World!
```

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

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

do/while迴圈敘述

- do/while和while迴圈敘述的不同處是在迴圈結尾檢查結束條件
- do/while迴圈的程式區塊至少會執行一次

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

跳出迴圈-- break

- 強迫終止迴圈的執行或繼續執行迴圈
- **break**關鍵字：當某些條件成立時，強迫終止迴圈的執行

```
if (number == null || number == target)  
    break;
```

繼續迴圈-- continue

- 可以馬上繼續下一次迴圈的執行
- 不會執行程式區塊中位在continue關鍵字之後的程式碼

```
if (number != target) {  
    document.write(number + "太小<br/>");  
    continue;  
}
```

JavaScript的巢狀迴圈

- 巢狀迴圈是指在迴圈之中擁有其他迴圈
- JavaScript的巢狀迴圈可以有很多層

```
for (i=1;i<=9;i++) {
```

```
.....
```

```
j = 1;
```

```
while (j <= 9) {
```

```
.....
```

```
j++;
```

```
}
```

```
}
```

練習2

□ 請使用for 和while迴圈建立兩層巢狀迴圈，以顯示九九乘法表

九九乘法表

	1	2	3	4	5	6	7	8	9
1	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	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	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	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	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	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	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	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	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