

選擇題

1. 下列哪個函數不能用於「預約程式碼的執行」或「取消預約程式碼的執行」？

- (1) setInterval (2) setInterval
(3) setTimeout (4) clearTimeout

2. 有一 JavaScript 程式碼片段如下：

```
<script>
function myFunc(n){
    if (n<=1)
        return(n);
    return(n*myFunc(n-1));
}
alert(myFunc(1.5));
</script>
```

請問上述程式碼執行後，會發生什麼事？

- (1) 跳出警告視窗，並顯示 1.5
(2) 跳出警告視窗，並顯示 0.75
(3) 跳出警告視窗，並顯示 undefined
(4) 進入無窮迴圈，網頁當掉，沒有回應
3. 如果我要做一個時鐘或是需要計時，最有可能用到下面哪個函數？
- (1) Date.clock(); (2) setTimeout();
(3) setTimeEnd(); (4) Timer.clock();
4. 下面的程式執行完後，x 和 y 的值分別為多少？

```
<script>
function testScope() {
    var x=5;
    y=5;
}
x = 10;
y = 10;
testScope();
</script>
```

- (1) $x = 10, y = 5$ (2) $x = 5, y = 5$
(3) $x = 10, y = 10$ (4) $x = 5, y = 10$
5. 若要重複使用 JavaScript 的函數，並將其寫成一個 "file.js" 的檔案，可用下列那個標籤及屬性以 Client-side include 的方式加入程式碼？

- (1) `<script jsFile=file.js></script>`
(2) `<script include=file.js></script>`
(3) `<script src=file.js></script>`
(4) `<script source=file.js></script>`

6. 有一 JavaScript 程式碼片段如下：

```
<script>
function myFunc(num){
    if (num <= 1)
        return(num);
    return(num*myFunc(num-1));
}
alert(myFunc(3));
</script>
```

請問上述程式碼執行後，會發生什麼事？

- (1) 跳出警告視窗，並顯示 1
(2) 跳出警告視窗，並顯示 3
(3) 跳出警告視窗，並顯示 6
(4) 進入無窮迴圈，網頁當掉，沒有回應
7. 在 HTML 開頭有一段 JavaScript 碼如下：

```
<script>
abc = 3;
function ShowABC(abc) {
    alert(abc);
}
</script>
```

請問在 HTML 其他位置呼叫 ShowABC(5) 時，跳出的 alert 視窗裡顯示的是何值？

- (1) 3 (2) 5
(3) undefined (4) NaN
8. 有關 JavaScript 變數的描述，下列何者有誤？
- (1) 若有需要，全域變數（Global Variables）可用 `global` 來進行宣告
 - (2) 區域變數（Local Variables）需用 `var` 來進行宣告
 - (3) 未宣告而直接設定的變數，均視為全域變數
 - (4) 全域變數的值可能在某個函數中被同名的區域變數所暫時遮蔽

簡答題

1. 請寫一段用戶端的 JavaScript 函數，其功能為使用非遞迴的方式來計算正整數 n 的階乘。
2. 請寫一段用戶端的 JavaScript 函數，其功能為使用遞迴的方式來計算正整數 n 的階乘。
3. 若有一個 JavaScript 函數，名稱是 `abc`，請問我們如何在此函數內取得輸入參數的個數？如何取得第一個參數？第二個參數？
4. 請問 `setTimeout()` 和 `clearTimeout()` 的功能為何？請舉實例說明。
5. 請問 `setInterval()` 和 `clearInterval()` 的功能為何？請舉實例說明。

程式題

請使用本章所學到 JavaScript 有關函數的程式技巧來完成下列作業：

1. (*) 圓的周長和面積之一：請寫一個網頁 `circleArea.htm`，包含一個連結「圓的周長和面積」，具有下列功能：
 - 當你按下此連結時，會跳出一個提示視窗，要求你輸入一個正數，做為一個圓的半徑。

- 此程式會算出此圓的周長和面積，並將結果顯示在網頁上面。（請用兩個函數來完成這個作業。）

（提示：你可以使用 `Math.PI` 來代表圓週率，會比 3.1415926 來的精準。）

2. (*) 圓的周長和面積之二：同前一小題，但請用一個函數來完成，此函數可以傳回一個陣列，包含兩個元素，分別是圓的周長和面積。（此網頁名稱是 `circleArea2.htm`。）

3. (**) 計算 Fibonacci 數列的遞迴函數：請寫一個網頁 `FiboRecursive.htm`，包含一個遞迴函數 `fibonacci(n)`，可用來計算第 n 項的 Fibonacci 數列，此數列的定義如下：

`fibonacci(0)=0`

`fibonacci(1)=1`

`fibonacci(n)=fibonacci(n-1)+fibonacci(n-2)`，當 n 大於或等於 2

請呼叫此函數，並在網頁列出從 $n = 0$ 到 $n = 20$ 的 `fibonacci(n)` 值。

4. (**) 計算 Fibonacci 數列的非遞迴函數：請重複上題，寫一個網頁 `FiboForLoop.htm`，但改用迴圈方式（非遞迴）的函數來完成。

（提示：有兩種方式來完成非遞迴的 Fibonacci 函數：

- 你可以在函數內宣告一個陣列，以便儲存 `fibonacci[0]`, `fibonacci[1]`, `fibonacci[2]` 等等的值，以便直接取用而不必重新算起。
- 你可以直接使用 Fibonacci 數列在第 n 項的數學式。（修過離散數學的同學都應該知道吧？）

5. (***) 計算時間比較：以「遞迴方式」和「迴圈方式」來產生 Fibonacci 數列：本題包含前面兩題。

- a. 請寫一個函數定義檔 `fibonacci.js`，裡面包含兩個函數，分別是遞迴函數 `fibonacciRecursive()` 和非遞迴函數（使用迴圈）`fibonacciForloop()`。
- b. 請寫一個網頁 `fibonacciSpeedTest.htm`，分別呼叫此函數，並進行計時，最後在網頁列出從 $n = 20$ 到 $n = 30$ 時，計算 `fibonacciRecursive(n)` 和 `fibonacciForloop(n)` 所花的時間，所列出的表格格式如下：

n	遞迴方式	迴圈方式
20	[計算 fiboRecursive(20)所花的時間]	[計算 fiboForloop(20)所花的時間]
21	[計算 fiboRecursive(21)所花的時間]	[計算 fiboForloop(21)所花的時間]
.	.	.
.	.	.
.	.	.
30	[計算 fiboRecursive(30)所花的時間]	[計算 fiboForloop(30)所花的時間]

- c. 你將會發現，電腦在計算 fiboForloop(n)所花的計算時間會比 fiboRecursive(n)少了很多，你能解釋原因嗎？
6. (***) 計算時間比較：以「遞迴方式」和「迴圈方式」來產生組合個數：一般組合個數是以 C_n^m 來表示，代表「由 m 個物件中，任意取出 n 個物件的所有可能情況個數」，其數學解析式為：

$$C_n^m = m! / [(m-n)!n!]$$

若不使用數學解析式，我們也可以使用下列遞迴式來表示：

$$C_n^m = C_{n-1}^{m-1} + C_n^{m-1}$$

(還記得此遞迴式所代表的意義吧？) 請重複上題，但改成組合數的運算，而列表則改成由 $m = 10 \sim 20$ ，而對應的 n 則是 m 的一半(小數不計)。(提醒：請特別注意邊界條件的指定。)

7. (*) 丟出去可回收的手榴彈之一：請寫一個網頁 `avoidableGrenade01.htm`，包含一個按鈕，功能如下：
- 按鈕表面的文字是「丟手榴彈」。
 - 按下按鈕後，文字變成「按下代表後悔，不然 5 秒內引爆...」
 - 若使用者在 5 秒內按下按鈕，則沒事，按鈕文字恢復成「丟手榴彈」。
 - 若使用者未在 5 秒內按下按鈕，則引爆手榴彈（跳出「轟!!!」的警告視窗），按鈕文字恢復成「丟手榴彈」。
- （提示：使用 `setTimeout()` 和 `clearTimeout()`，程式碼會比較簡潔。）
8. (**) 丟出去可回收的手榴彈之二：此題類似前一題，請寫一個網頁 `avoidableGrenade02.htm`，包含一個按鈕，功能如下：
- 按鈕表面的文字是「丟手榴彈」。
 - 按下按鈕後，文字變成「按下代表後悔，不然 5 秒內引爆...」，但此文字會隨著時間而變化，秒數會變成「5 秒」、「4 秒」、「3 秒」等等，直到引爆或取消。
 - 若使用者在 5 秒內按下按鈕，則沒事，按鈕文字恢復成「丟手榴彈」。
 - 若使用者未在 5 秒內按下按鈕，則引爆手榴彈（跳出「轟!!!」的警告視窗），按鈕文字恢復成「丟手榴彈」。
9. (**) 左右跑馬燈：請寫一個網頁 `movingTextLeftRight.htm`，包含類似下列的表單：

曾經滄海難為水，除卻巫山不是雲。取次花叢懶回顧，半絳		
<===	STOP	===>

功能如下：

- 按下「<===」按鈕，跑馬燈文字往左邊跑。
- 按下「===>」按鈕，跑馬燈文字往右邊跑。
- 按下「STOP」按鈕，跑馬燈文字靜止不動。

（提示：使用 `setInterval()` 和 `clearInterval()`，程式碼會比較簡潔。）