

## 選擇題

1. 假設有一個函數檔案 `utility.js`，包含 JavaScript 的各種函數，已經可以用於 Client-side Include，請問在伺服器端的 ASP 程式碼，如何引用此檔案？
  - (1) `<script runat=server language=javascript src="utility.js"></script>`
  - (2) `<script src="utility.js"></script>`
  - (3) `<!--#include file="utility.js"-->`
  - (4) `<!--#include source="utility.js"-->`

## 簡答題

1. 請使用一個實例，來說明如何在一個 ASP 的網頁中呼叫一個 JavaScript 的遞迴函數。
2. 請使用一個實例，來說明如何在一個 ASP 的網頁中呼叫一個具有可變參數個數的 JavaScript 函數。
3. 請說明兩種方法，可將常用的 JavaScript 函式寫在一個檔案內，以便被不同的 ASP 網頁呼叫。（請舉一個實例來說明。）

## 程式題

請使用本章所學到 Server-side JScript 有關函數的程式技巧來完成下列作業：

1. (\*\*) 計算 Fibonacci 數列的遞迴函數：請寫一個網頁 `FiboRecursive.asp`，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：  
$$\text{fibonacci}(0)=0$$
$$\text{fibonacci}(1)=1$$
$$\text{fibonacci}(n)=\text{fibonacci}(n-1)+\text{fibonacci}(n-2), \text{ 當 } n \text{ 大於或等於 } 2$$

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

2. (\*\*) 計算 Fibonacci 數列的非遞迴函數：請重複上題，寫一個網頁 `FiboForLoop.asp`，但改用迴圈方式（非遞迴）的函數來完成。（提示：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。）
3. (\*\*\*) 計算 Fibonacci 數列的遞迴函數，並計算在用戶端和伺服器端所花的計算時間：請寫一個網頁 `fibonacciSpeedTest.asp`，裡面包含計算 Fibonacci 數列的遞迴函數（請見第一題），並分別呼叫此函數，在伺服器端和用戶端分別計時，最後在網頁列出從  $n = 10$  到  $n = 20$  時，在伺服器端和用戶端計算此數列所花的時間，所列出的表格格式如下：

n	伺服器端的計算時間	用戶端的計算時間
10	[伺服器計算 fibonacci(10)所花的時間]	[用戶端計算 fibonacci(10)所花的時間]
11	[伺服器計算 fibonacci(11)所花的時間]	[用戶端計算 fibonacci(11)所花的時間]
.	.	.
.	.	.
.	.	.
20	[伺服器計算 fibonacci(20)所花的時間]	[用戶端計算 fibonacci(20)所花的時間]

- （提示：你必須寫一個檔案 `fibonacci.js`，包含 `fibonacci(n)` 函數，並同時在伺服器端和用戶端引用這一個檔案。）
4. (\*\*\*) 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.asp`，使用 SSI (Server-Side Include) 的方式來引入上述檔案，並分別呼叫此函數，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？