

# 函式

## 12-1 函式的用途

- 函式具有可重複使用性。
- 在加上函式後，程式碼會變得更精簡。
- 在加上函式後，程式碼的可讀性會提高。
- 在將程式碼拆成幾個函式後，寫起來會比較輕鬆，而且程式碼的邏輯和正確性都會提高，容易偵錯、修改與維護。

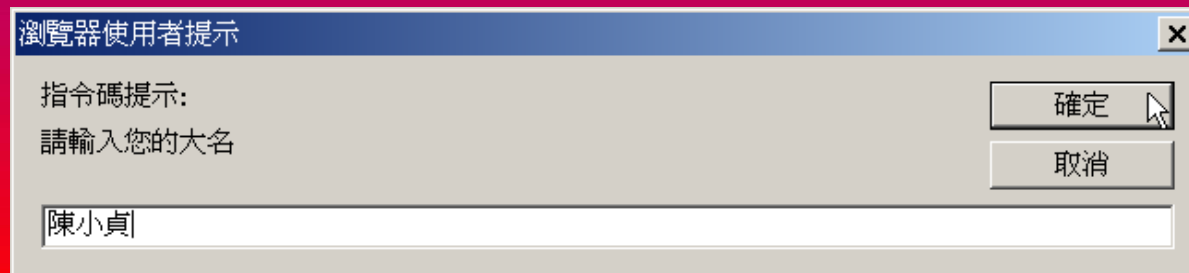
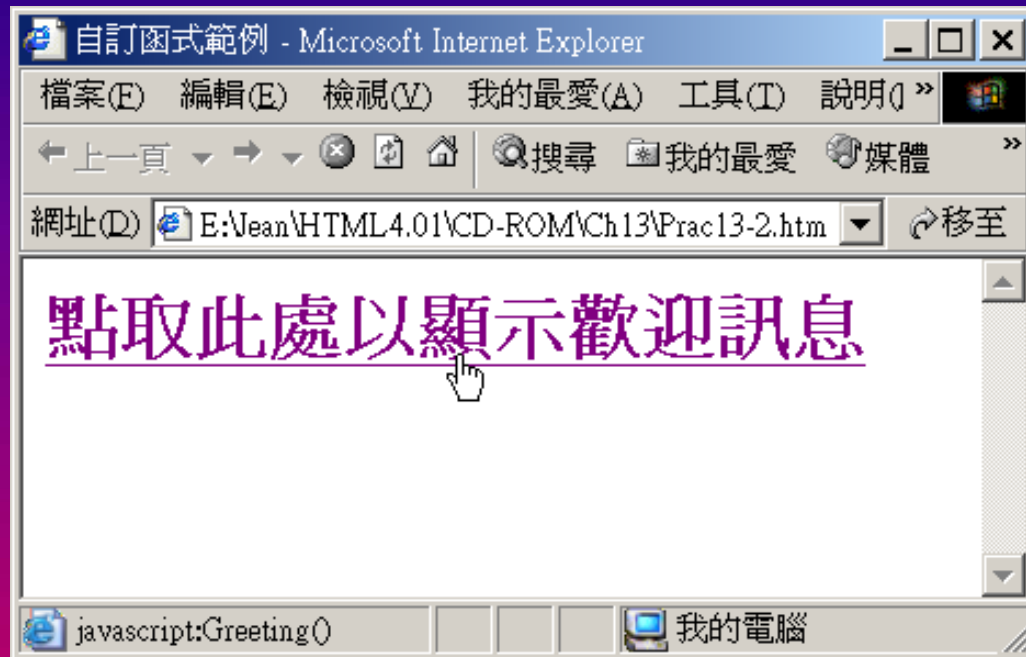
## 12-2 使用者自訂函式

```
function function_name([argumentlist])  
{  
  statements;  
  [return;|return value];  
  [statements];  
}
```

```
01:<HTML>
02: <HEAD>
03:  <TITLE>自訂函式範例</TITLE>
04:  <SCRIPT LANGUAGE="javascript">
05:    function Greeting()    //定義函式
06:    {
07:      var UserName = prompt("請輸入您的大名", "");
08:      alert(UserName + "您好！歡迎光臨！");
09:    }
10:    Greeting();            //呼叫函式
11:  </SCRIPT>
12: </HEAD>
13: <BODY>
14: </BODY>
15:</HTML>
```

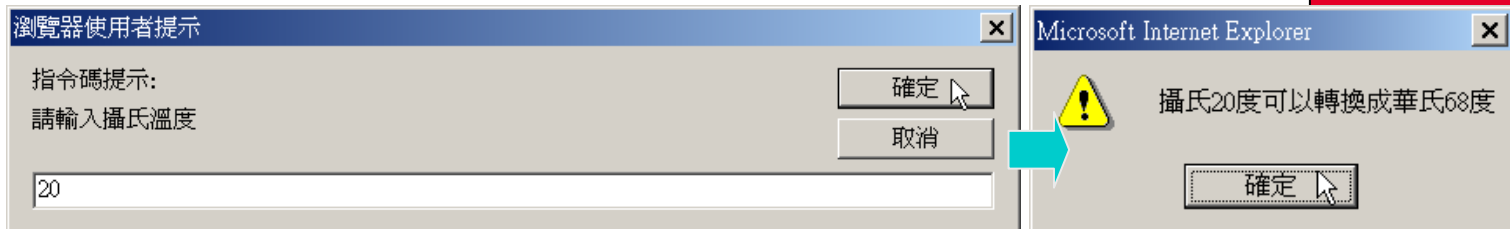


```
01:<HTML>
02: <HEAD>
03:   <TITLE>自訂函式範例</TITLE>
04:   <SCRIPT LANGUAGE="javascript">
05:     function Greeting()
06:     {
07:       var UserName = prompt("請輸入您的大名", "");
08:       alert(UserName + "您好！歡迎光臨！");
09:     }
10:   </SCRIPT>
11: </HEAD>
12: <BODY>
13:   <H1><A HREF="javascript:Greeting()">點取此處以顯示歡
    迎訊息</A></H1>
14: </BODY>
15:</HTML>
```



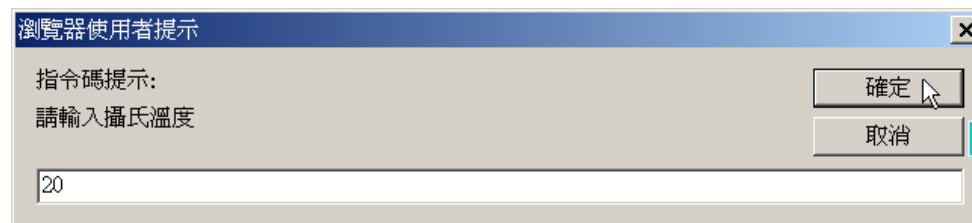
# 12-2-1 函式的參數

```
01:<HTML>
02: <HEAD>
03:  <TITLE>自訂函式範例</TITLE>
04:  <SCRIPT LANGUAGE="javascript">
05:    function Convert2F(DegreeC)
06:    {
07:      var DegreeF = DegreeC * 1.8 + 32;
08:      alert("攝氏" + DegreeC + "度可以轉換為華氏" + DegreeF + "度");
09:    }
10:    var Temperature = prompt("請輸入攝氏溫度", "");
11:    Convert2F(Temperature);
12:  </SCRIPT>
13: </HEAD>
14: <BODY>
15: </BODY>
16:</HTML>
```



## 12-2-2 函式的傳回值

```
01:<HTML>
02: <HEAD>
03:  <TITLE>自訂函式範例</TITLE>
04:  <SCRIPT LANGUAGE="javascript">
05:    function Convert2F(DegreeC)
06:    {
07:      return DegreeF = DegreeC * 1.8 + 32;
08:    }
09:    var Temperature = prompt("請輸入攝氏溫度", "");
10:    var Result = Convert2F(Temperature);
11:    alert("攝氏" + Temperature + "度可以轉換為華氏" + Result + "度");
12:  </SCRIPT>
13: </HEAD>
14: <BODY>
15: </BODY>
16:</HTML>
```





## 12-3 區域變數V.S. 全域變數

```
01:<HTML>
02: <HEAD>
03:  <TITLE>自訂函式範例</TITLE>
04:  <SCRIPT LANGUAGE="javascript">
05:    var Msg;                                //宣告一個全域變數Msg
06:    Msg = "Hello, This is outside of Func1().";    //設定全域變數Msg
    的值
07:    alert(Msg);                            //顯示全域變數Msg
    的值
08:    Func1();                                //呼叫Func1() 方法
09:    alert(Msg);                            //顯示全域變數Msg
    的值
10:
```

## 12-3 區域變數V.S. 全域變數

```
11: function Func1()  
12: {  
13:     var Msg; //宣告另一個同名的  
區域變數Msg  
14:     Msg = "Hello, This is inside of Func1()."; //設定區域變數Msg  
的值  
15:     alert(Msg); //顯示區域變數Msg  
的值  
16: }  
17: </SCRIPT>  
18: </HEAD>  
19: <BODY>  
20: </BODY>  
21:</HTML>
```



```
01:<HTML>
02: <HEAD>
03:  <TITLE>自訂函式範例</TITLE>
04:  <SCRIPT LANGUAGE="javascript">
05:    var Msg;                                //宣告一個全域變數Msg
06:    Msg = "Hello, This is outside of Func1().";//設定全域變數Msg的值
07:    alert(Msg);                             //顯示全域變數Msg的
    值
08:    Func1();                                //呼叫Func1() 方法
09:    alert(Msg);                             //顯示全域變數Msg的值
10:    function Func1()
11:    {
12:      Msg = "Hello, This is inside of Func1().";//設定全域變數Msg的值
13:      alert(Msg);                           //顯示全域變數Msg的值
14:    }
15:  </SCRIPT>
16: </HEAD>
17: <BODY>
18: </BODY>
19:</HTML>
```



## 12-4 JavaScript內建函式

### 12-4-1 encodeURIComponent()、decodeURI()

- encodeURIComponent() 會將英文字母、數字及!#\$&'()\*+,-./:;=?@\_~之外的字元加以編碼，例如encodeURIComponent("Hello 123!") 會傳回Hello%20123!。
- decodeURI() 可以將被encodeURIComponent() 編碼過的資料加以解碼，例如decodeURI("Hello%20123!") 會傳回Hello 123!。

## 12-4-2 encodeURIComponent()、decodeURIComponent()

- encodeURIComponent() 和 encodeURI() 類似，不同的是它會將英文字母、數字及 ! ' ( ) \* - . \_ ~ 之外的字元加以編碼，例如 encodeURI("http://") 會傳回 http://，encodeURIComponent("http://") 會傳回 http%3A%2F%2F。
- decodeURIComponent() 可以將被 encodeURIComponent() 編碼過的資料加以解碼，例如 decodeURIComponent("http%3A%2F%2F") 會傳回 http://。

## 12-4-3 eval()

eval() 函式的用途是將字串當成 JavaScript 程式碼執行，例如：

```
eval("alert('Hello World!');");
```





## 12-4-4 isFinite()、isNaN()

- isFinite() 用來判斷參數是否為有限值
- isNaN() 用來判斷參數是否為NaN (Not a Number)

## 12-4-5 Number()、parseFloat()、parseInt()

- Number() 用來將參數轉換為數字
- parseInt() 用來將參數轉換為整數
- parseFloat() 用來將參數轉換為浮點數
  - Number("10.5") + Number("8"); //傳回數字18.5
  - Number("abc"); //傳回NaN
  - parseInt(10.5); //傳回10
  - parseInt("10.5"); //傳回10
  - parseInt("-7.56"); //傳回-7
  - parseFloat(10.5); //傳回10.5
  - parseFloat("10.5"); //傳回10.5
  - parseFloat("-7.56"); //傳回-7.56

## 12-5 函式庫

```
function Greeting()  
{  
  var UserName = prompt("請輸入您的大名", "");  
  alert(UserName + "您好！歡迎光臨！");  
}
```

```
<HTML>
  <HEAD>
    <TITLE>自訂函式範例</TITLE>
    <SCRIPT SRC="MyLibrary.js"></SCRIPT>
    <SCRIPT LANGUAGE="javascript">
      Greeting();
    </SCRIPT>
  </HEAD>
  <BODY>
  </BODY>
</HTML>
```

