



JavaScript 精選16堂課

網頁程式設計實作



第一部分：JavaScript精要

- 第一堂課：認識JavaScript
- 第二堂課：JavaScript基礎語法
- 第三堂課：程式控制結構
- 第四堂課：JavaScript內建標準物件
- 第五堂課：集合物件
- 第六堂課：函式與作用域
- 第七堂課：物件、方法與屬性
- 第八堂課：RegExp物件
- 第九堂課：非同步與事件循環(Event loop)

第一堂課：認識JavaScript

1-1 JavaScript特色與用途

1-1-1 JavaScript基本觀念

- Javascript是一種直譯式(Interpret)的描述語言
- 具有跨平台、物件導向、輕量的特性
- 通常會與其他應用程式搭配使用，最廣為人知的當屬Web程式的應用。例如：JavaScript與HTML及CSS搭配製作具有互動效果的網頁
- JavaScript 所採用的標準是 ECMAScript
- 瀏覽器較能完整支援的標準為ECMAScript2014，第五版(簡稱ES5)以及ECMAScript2015，第六版(簡稱ES6)
- 最新的版本是2018年9月釋出的ECMAScript 2018，第9版(簡稱ES9)

1-1-2 JavaScript的用途

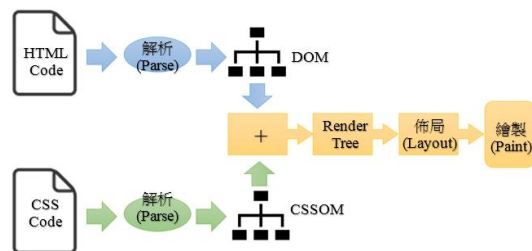
- 操作HTML DOM
- 網頁遊戲
 - Sumon
 - 網址：<https://sumonhtml5.ludei.com/>
 - Emberwind
 - 網址：<http://operasoftware.github.io/Emberwind/>
- 操作HTML5前端資料儲存
- Node.js後端平台

1-2 設置JavaScript開發環境

- 1-2-1 JavaScript運行環境
 - 傳統的JavaScript運行環境只能夠在前端(用戶端)運行，Node.js的出現讓JavaScript也能夠在後端(伺服器端)執行
- JavaScript的主要核心：
 - ECMAScript：定義程式語法、流程控制、資料型別、物件與函數、錯誤處理機制等基本語法
 - DOM API：用來存取及改變網頁文件物件結構與內容

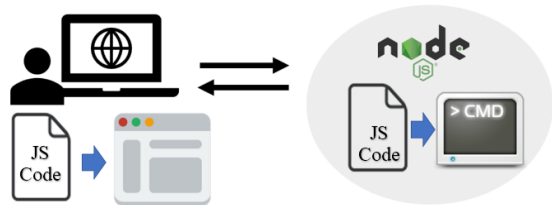
JavaScript在用戶端執行

- 將HTML碼與CSS碼交給渲染引擎(Render Engine)處理
- HTML碼解析並建構DOM樹狀結構(文件物件模型，Document Object Model)
- CSS碼解析並建構CSSOM樹狀結構(CSS物件模型，CSS Object Model)
- 依照順序組合成渲染樹狀結構(Render Tree)
- 透過Layout()方法依照每個節點的坐標位置與大小來安排版面
- 執行Paint()方法在瀏覽器繪製出網頁



JavaScript在後端執行

- Node.js使用Google的V8引擎
- Node.js官方網站網址：
<https://Node.js.org/en/>
- LTS(Long Term Support)版本通常是比較穩定的版本
- Node.js提供了一個類似終端機模式的REPL環境(Read Eval Print Loop，稱為交互式開發環境)



REPL環境(Read Eval Print Loop，交互式開發環境)

- 在命令提示字元視窗(CMD)輸入「node」按下Enter鍵，出現REPL的提示字元(>)，表示已經進入REPL環境

```
C:\Users\Administrator>node
> _
```

- 直接輸入JavaScript程式碼「console.log("Hello World");」

```
C:\Users\Administrator>node
> console.log("Hello World");
Hello World
undefined
>
```

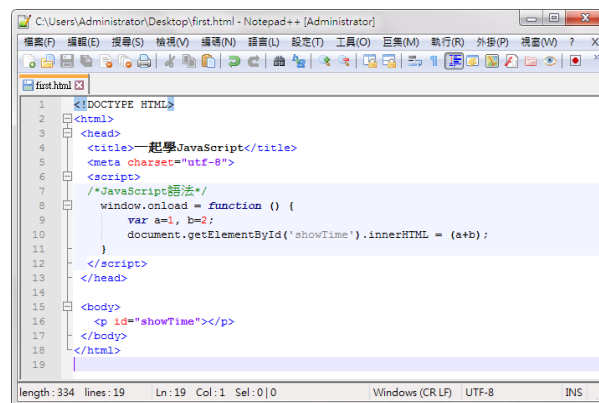
- 離開REPL環境有兩種方式：
 - 輸入「.exit」
 - 按下Ctrl+D

1-2-2 如何選擇文字編輯器

- 純文字編輯器：EditPlus、NotePad++、PSPad、UltraEdit、Visual Studio Code等等，基本特色：
 - 編輯功能
 - 程式碼著色
 - 顯示行號
- IDE工具：WebStorm、Visual Studio、Eclipse，除基本功能外，通常會額外提供下列功能：
 - 版本控制
 - 指令自動完成
 - 程式碼檢查
 - 除錯(Debug)
 - 程式碼自動縮排

純文字編輯器建議具備的功能

- 具備選取、剪下、複製、貼上、搜尋等基本功能
- 支援多次復原和恢復
- 語法著色
- 結構檢視
- 顯示行號



1-2-3 純文字編輯器NotePad++

- 語法著色及語法摺疊功能
- 自動完成功能(Auto-completion)
- 自動補齊功能
- 支援同時編輯多重檔案
- 支援多重視窗同步編輯
- 支援 PCRE (Perl Compatible Regular Expression) 搜尋及取代
- 提供程式碼區放大與縮小功能
- 高亮度括號及縮排輔助
- 巨集

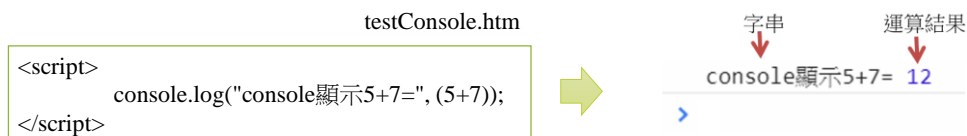


1-2-4 瀏覽器主控台console

- 撰寫前端JavaScript程式，開發者最常利用的瀏覽器工具應該就屬開發者工具Console(中文稱主控台)
- 各家瀏覽器都有自己的Console，操作方式不盡相同，但基本功能大同小異
- 底下介紹google chrome的DevTools Console

Google DevTools Console

- 打開Chrome瀏覽器並開啟書附光碟ch01/testConsole.htm，按下「F12」鍵，瀏覽器右邊會顯示DevTools主控台的console面板

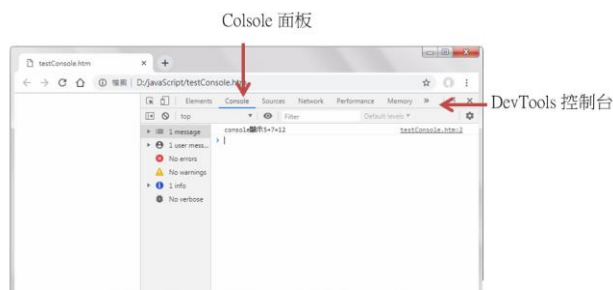


- 也可以寫成(`"console顯示5+7=" + (5+7)`)輸出字串


`console顯示5+7=12` ← 字串

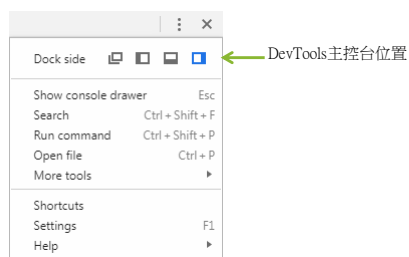
主控台console

- Console是操作主控台console物件的API，提供許多方法供我們使用
`console.log()`是其中一個方法，功能是輸出一些訊息到主控台



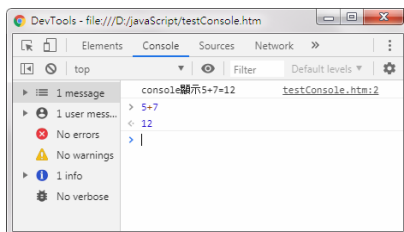
主控台的位置

- DevTools主控台的右上方  可以選擇主控台的位置，從左至右依序是浮動、置於左方、置於下方、置於右方



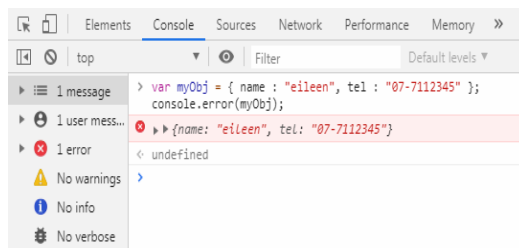
Console直接執行JS

- Console不僅可以輸出JavaScript的訊息，也可以直接執行JavaScript程式碼
- 只要在console面板按一下滑鼠左鍵，這時會出現游標
- 輸入JS程式碼，輸入完成之後，按下Enter鍵就會執程式



Console物件方法(一)

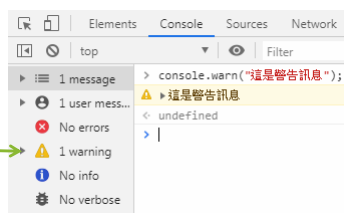
- `assert()` `assert(assertion,錯誤訊息)`
 - `assertion`是一種邏輯判斷式，結果只有真(True)跟假(False)
- `error()` `console.error(message)`
 - `error()`方法會輸出錯誤訊息到主控台，括號內是放置要顯示的訊息，可以是字串或是物件



Console物件方法(二)

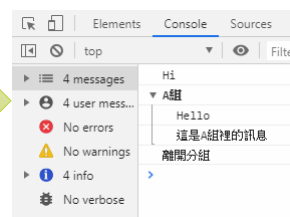
- `warn()` `console.warn(message)`
- `clear()` `console.clear();`
- `count()`
 - `Count()` `console.count(label);` 次數，括號內可以放置要辨識的標籤，不加標籤則以default顯示
- `group()` 與 `groupEnd()`

warning類型 →



`console.group(label)` //開始分組
`console.groupEnd()` //結束分組

```
console.log("Hi");
console.group("A組");
console.log("Hello");
console.log("這是A組裡的訊息");
console.groupEnd();
console.log("離開分組");
```

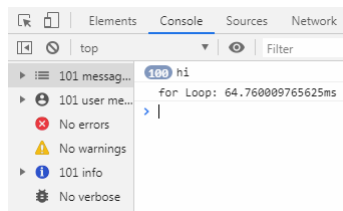


Console物件方法(三)

- time()與timeEnd()

```
console.time(label) //開始計時  
console.timeEnd(label) //結束計時
```

```
console.time("for Loop"); //開始計時  
for (i = 0; i < 100; i++) {  
  console.log("hi")  
}  
console.timeEnd("for Loop"); //結束計時
```



- 儲存Console面板的訊息

- Console面板空白處按右鍵執行「save as...」

