

總結過去幾十年的 JavaScript 開發，可以把它粗略地分為四個主要時代：

1. 前世
2. 第一代框架(jQuery)
3. 以元件為中心的視圖層
4. 全棧式框架 (← 這也是我們當前所處的位置)

前端領域是變得簡單還是越來越難？

「每 18 至 24 個月，前端都會難一倍」

赫門 《前端服務化之路》 @ 2015 shenJS

是變得簡單還是越來越難要看對於技術的掌握度以及專案的規模來決定。

如果只能用一句話來解釋，就是前端框架/工具庫的發展方向，實際上是「讓簡單的專案變得複雜，讓複雜專案的開發變得單純」。

前端框架/工具庫的分類

大致上來說，前端框架/工具庫分做這幾種：

- 以操作 DOM 為主：

像大家都很熟悉的 jQuery、YUI (我們懷念它)、或是早期的 PrototypeJS (不是 JavaScript 原生的那個 prototype)、Dojo 等都屬於這一類型。

最近幾年才進入前端領域的朋友，可能很難想像當年 jQuery 提出可以讓開發者直接以 CSS Selector 來操作 DOM 的做法。當然 jQuery 會紅到現在其實也不只這點，還有各種其他因素就是。

此類工具主打的就是對 DOM 的操作變得更簡便。優點就是上手容易，操作直覺，拿到 DOM 之後就直接對它做事。早期大約十幾年前的主流工具庫多屬於這種類型。

- 整合 Web UI、處理圖形化介面為主：

這類的工具庫通常屬於是對前一個「以操作 DOM 為主」的再擴充。

主要想解決的目標的是把包裝好的各種圖形化介面直接拿來使用。

像 Bootstrap、jQuery UI、Kendo UI 等這種已經幫你把幾個常用的圖形化介面都封裝好，開發者只需要準備好容器的節點，幾乎只要一初始化就可以直接使用。當然這種工具也是兩面刃，優點是裝了就可以直接用，但是如果要高度客製化、擴充跟改寫，如果函式庫本身沒有提供相關 api 給開發者，那麼改起來恐怕比自己刻還累。

JavaScript 基礎

JavaScript 是一個可以幫您在網站裡加入互動功能的程式語言（舉例來說，一個遊戲可能會在按鈕按下或資料被輸入表單內時回應、動態更改樣式、以及展示動畫等）。這篇文章會幫助您踏上學習這個令人興奮的語言的旅程，並展示她可以實現的所有可能。

所以 JavaScript 到底是什麼？

[JavaScript](#) 是一個成熟的動態程式語言，應用於 [HTML](#) 文件（document）上時，就可以為網頁提供動態的互動功能。JavaScript 是由 Mozilla project、Mozilla Foundation 和 Mozilla Corporation 的創辦人 Brendan Eich 所發明的。

你可以用 JavaScript 實現許多事情。你可以先從簡單的特性開始，如跑馬燈、相簿、動態版型、回應按鈕點擊等。在你熟悉的這個程式語言以後，甚至可以製作遊戲、2D 平面以及立體的圖像、資料庫系統等等的應用！

JavaScript 本身非常的簡潔，卻也充滿彈性，開發者們已經以 JavaScript 核心為基礎為她撰寫了相當多的工具，讓各位可以感到事半功倍。這些工具包括：

- 內建在瀏覽器內的應用程式介面（Application Programming Interfaces，[APIs](#)）提供了多樣化的功能，像是動態產生 HTML 以及設定 CSS 樣式、擷取以及處理從使用者的網路攝影機錄下的影像、製作立體圖形或是聲音樣本。
- 第三方 API 允許開發者將他們的網頁與其他如 Twitter 或 Facebook 提供的內容合併在一起。
- 第三方框架和函式庫允許您將這些元件套用在您的 HTML 文件內，讓您可以迅速地建立網頁或應用程式。

「Hello world」範例程式

前面所述的功能聽起來令人興奮，而她也的確符合這樣的期待— JavaScript 是眾多令人感到興奮的網路科技之一，您會因為選擇利用她來製作網頁而進入一個嶄新且充滿創意及力量的次元。

但無論如何，要讓 JavaScript 跟 HTML 和 CSS 合作無間的話，可能還要費一些功夫。現在您將會從一些細小的地方開始著手，接著一步步地往前進。首先，

我們將會向您展示如何將一些基本的 JavaScript 給加入您的頁面中，並且打造一個*「hello world!」的範例。

警告：如果您還沒有學習過先前的課程，[請您下載這個範例程式碼](#)，並以此開始練習。

1. 首先，進入您測試網頁的資料夾中，並建立一個名為main.js的檔案，再將她存放於scripts資料夾內。
2. 接著，開啟index.html檔案，並在</body>這個結束標籤之前的位置，使用一行新的空間來輸入以下的元素：
3. <script src="scripts/main.js"></script>
4. 我們做的事情，基本上跟新增一個 CSS 的<link>元素是相同的概念:

我們將 JavaScript 給導入這個頁面中，讓她來影響 HTML（以及 CSS、還有任何頁面上的東西）。

5. 再來我們把以下的程式碼新增到main.js檔案內：

```
var myHeading = document.querySelector('h1');  
myHeading.textContent = 'Hello world!';
```

6. 現在請您將修改過的 HTML 和 JavaScript 給存檔，再用瀏覽器讀取index.html。您應該會看到以下的內容：



備註：選擇將<script>元素放在接近 HTML 檔案底部的原因，是因為瀏覽器是依照程式碼存在檔案中的順序來讀取 HTML 檔案的。如果 JavaScript 先被瀏覽器讀取了，就無法啟動去影響之後的產生改變的 HTML 程式碼了。因此，把 Javascript 放在接近檔案底部的位置，通常都會是一個不錯的策略。

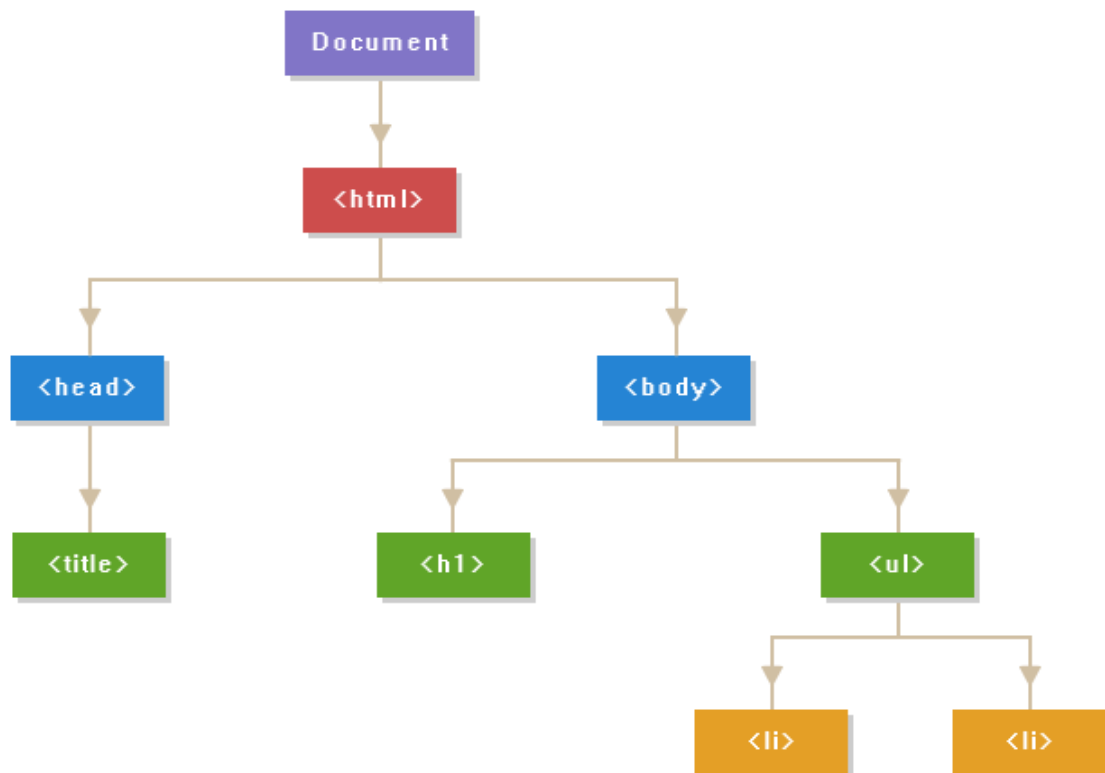
且也是有利於協同開發的程式人員找到 Javascript 的程式碼在哪裡，方便管理。

[發生什麼事了？](#)

所以您的標題文字已經被 JavaScript 修改成「Hello world!」了。我們先使用了一個叫做[querySelector\(\)](#)的函式來取得了我們標題參考（Reference），並且將她存在一個叫做myHeading的變數裡面。這跟我們在操作 CSS 時使用的選擇器是相似的。當您想要更動某個元素時，首先您要將她選取起來。

之後，我們將變數 myHeading 中 [innerHTML](#) 特性的值設為「Hello world!」。

備註：Both of the features you used above are parts of the[Document Object Model \(DOM\) API](#), which allows you to manipulate documents.



[定義變數](#)

你可以透過三種方式來定義變數：

- 透過保留字 [var](#) 來定義變數，舉例來說：var x = 42，這種方式可以用來定義區域以及全域變數。
- 直接指定一個值給該變數，例如：x = 42，這種方式只能定義全域變數，如果在方法外面使用該方法定義變數，嚴格模式裡會產生警告，該定義方式應該盡可能避免。
- 透過保留字 [let](#)，舉例來說：let y = 13，[let](#)可以用來定義區塊裡的區域變數。

JavaScript 的保留字

在 JavaScript 中不能使用這些保留字作為變量，標籤或函數名：

abstract	arguments	boolean	break	byte
case	catch	char	class*	const
continue	debugger	default	delete	do
double	else	enum*	eval	export*
extends*	false	final	finally	float
for	function	goto	if	implements
import*	in	instanceof	int	interface
let	long	native	new	null
package	private	protected	public	return
short	static	super*	switch	synchronized
this	throw	throws	transient	true
try	typeof	var	void	volatile
while	with	yield		

標有*的詞是新ECMAScript5



jQuery 是什麼？

jQuery 是一個 JavaScript 函式庫 (Javascript Framework)，主要目的就跟 Logo 上面寫的精神一樣：「write less, do more」，由於如果要直接用 Javascript 將所有功能實作出來，常常要費時費力且造成大量且難以閱讀的程式碼，更嚴重者，有於瀏覽器之間的差異，常常要針對不同瀏覽器寫不同的程式碼，讓程式碼維護的難度大為提升。

簡單說，jQuery 就是一個讓你加速網頁程式開發速度的工具，就是所謂的**敏捷式開發**。

這邊先用一段程式碼來讓大家了解 jQuery 的威力：

假設今天想要用 Javascript 讀取網頁中的單選按鈕「radio Button」，程式碼如下：

```
1 var checkValue;

2 var elements = document.getElementsByTagName('input');

3 for (var n = 0; n < elements.length; n++){

4 if (elements[n].type == 'radio' && elements[n].name == 'RadioGroup' &&
  elements[n].checked){

5 checkedValue = elements[n].value;

6 }
```

整個流程應該會是 (1) 找出所有 input 元素 (2) 當元素是 radio button 且屬於某個 group (3) 確認某個選項已經被勾選了 (4) 將勾選的值指派給 checkValue 變數

但如果使用 jQuery 的話，要怎樣完成上面的命令呢，只要一行

```
var checkValue = $('[name="radioGroup"]:checked').val();
```

其中的 `$()` 符號就是 jQuery 常用的選擇器，善用選擇器可以快速找到網頁當中的物件，所以 JQuery 是簡化繁鎖的 Javascript，但如果沒有 Javascript 的基礎，JQuery 語法就會像有字天書，讓人摸不著頭緒。

jQuery 標準語法格式:

```
$(document).ready(function(){  
    執行程式...  
});
```

這個語法簡化為:

```
$(function(){  
    執行程式...  
});
```

jQuery 基本選擇器:

1. 標籤選擇器:

\$('a')

功能相當於 Javascript 的 `getElementsByName()` 方法

例如:

```
Document..getElementsByName('a');
```

2. ID 選擇器:

\$('#myName')

功能相當於 Javascript 的 `getElementById()` 方法

例如:

```
Document..getElementById('myName');
```


3.類別選擇器:

`$('.myCSS')`

功能相當於 Javascript 的 `getElementsByName()` 方法

例如:

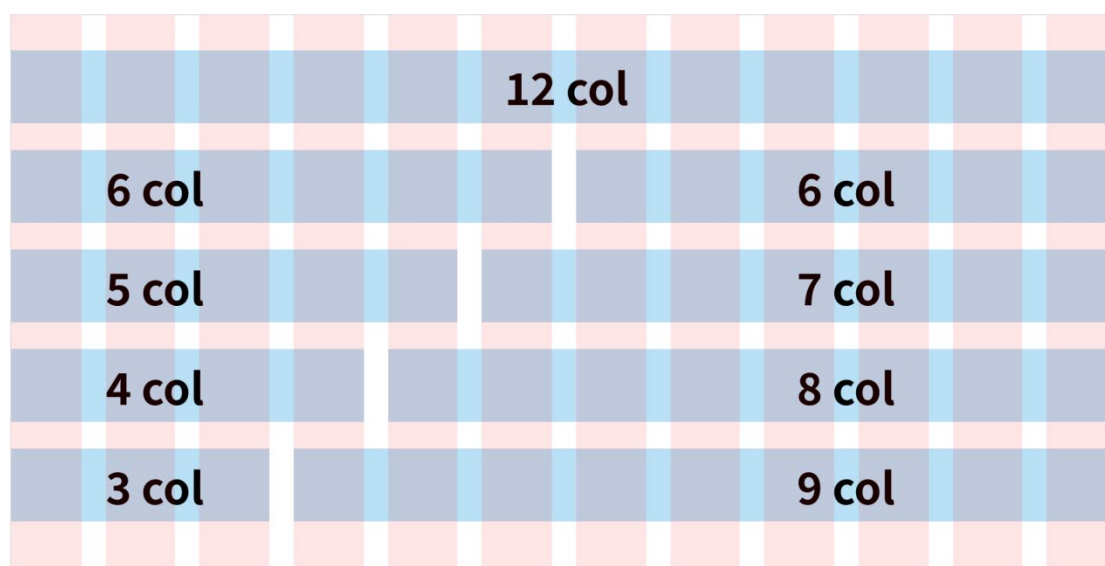
`Document..getElementsByName('myName');`

Bootstrap 網頁框架

Bootstrap 是一個由 HTML、CSS 和 JavaScript 寫成的前端框架，核心的設計目標是達成 RWD 響應式與行動優先，也就是讓你的網站排版可以自動適應螢幕大小。它預先做好一套網站的基礎建設，讓你能在框架的基礎上進行開發，不需要再去煩惱瑣碎的設定。

Grid system | 網格系統概述

Bootstrap 提供了一套響應式、行動裝置優先的網格系統，隨著螢幕 (viewport) 尺寸的增加，系統會自動分為最多 12 列。它是一種用於快速創建一致的佈局和有效地使用 HTML 和 CSS 的方法。



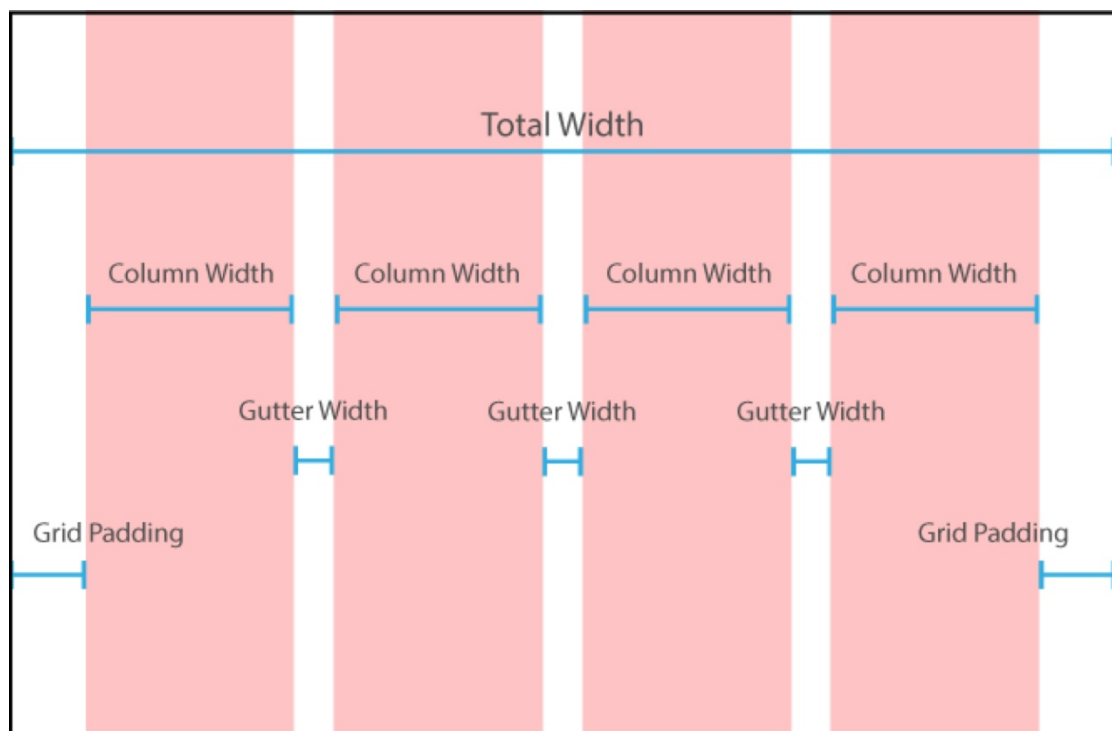
Media Queries 的分段點

1. /* 超小設備（手機，小於 768px） */
Mobile – xs (< 768px)/*
.col-xs-
2. /* 小型設備（平板電腦，768px 起） */
Tablet – sm (768~991px)
.col-sm-

3. /* 中型設備（筆記型電腦，**992px** 起） */
Desktop – md (992~1200px)
.col-md-
4. /* 大型設備（大台式電腦，**1200px** 起） */
Large Desktop - lg (>= 1200px)
.col-lg-

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
.container	100%	540px	720px	960px	1140px	1320px
.container-sm	100%	540px	720px	960px	1140px	1320px
.container-md	100%	100%	720px	960px	1140px	1320px
.container-lg	100%	100%	100%	960px	1140px	1320px
.container-xl	100%	100%	100%	100%	1140px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px
.container-fluid	100%	100%	100%	100%	100%	100%

- 格線系統關鍵字：**column(欄)**、**gutter(間距)**



Bootstrap 5 格線系統原理

- 排版三大重要觀念：`.container`、`.row`、`.col 1~12`
 - `.container`
 - `.container` 固定寬度
 - `.container-fluid` 滿版
 - `.row`
 - `no-gutters`：不需要 `gutters`
- 規則 1：`.col-` 的外層只能是 `.row`
- 規則 2：`.row` 裡面只能是 `.col-`
- 規則 3：網頁內容與元件請放在 `.col-` 裡面

常見錯誤，在 `.col` 增加寬度

常見錯誤，在格線系統調整左右 `margin` 與 `padding`

常見觀念：可以加上下 的 `margin` 與 `padding`

常見觀念：最外層至少補一個 `container`

常見觀念：整體格線邏輯是一致

範例

```
<div class="container">
  <div class="row">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
</div>
```

通用類別

間隔 (Spacing)

Bootstrap 包括各種簡寫響應式 `margin`、`padding` 和間隔的通用類別，用來修改一個元素的外觀。

Margin and padding

賦予一個縮寫 `class` 使 `margin` 或 `padding` 值在一個元素上或其中一個邊緣上 (包含響應式)。包含支援單一個邊緣屬性或全部邊緣、垂直邊緣、水平邊緣。`Class` 來自於 `Sass map`，範圍從 `.25rem` 到 `3rem`。

符號

如果是適用於所有從 `xs` 到 `xxl` 斷點的間隔通用類別，就不需加入中斷點縮寫。因為從 `min-width: 0` 以上開始都將應用這些類別，因此不受 `media query` 的約束。但若是針對其餘斷點，就需要包含中斷點縮寫。

對於 `xs` 使用固定格式 `{property}{sides}-{size}` 命名，對於 `sm`、`md`、`lg`、`xl` 和 `xxl`，使用格式 `{property}{sides}-{breakpoint}-{size}` 命名。

property 設定:

- `m` - 設定 `margin` 的類別
- `p` - 設定 `padding` 的類別

sides 設定:

- `t` - 設定 `margin-top` 或是 `padding-top` 的類別
- `b` - 設定 `margin-bottom` 或是 `padding-bottom` 的類別

- **s** - (start) 在 LTR 設定 **margin-left** 或是 **padding-left**，RTL 設定 **margin-right** 或是 **padding-right**
- **e** - (end) 在 LTR 設定 **margin-right** or **padding-right**，RTL 設定 **margin-left** 或是 **padding-left**
- **x** - 同時設定 ***-left** 和 ***-right**
- **y** - 時設定 ***-top** 和 ***-bottom**
- **blank** - 同時設定 **margin** 或 **padding** 在元素的四個邊緣

size 設定:

- **0** - 設定 **margin** 或是 **padding** 為 0
- **1** - (預設) 設定 **margin** 或是 **padding** 為 $\$spacer * .25$
- **2** - (預設) 設定 **margin** 或是 **padding** 為 $\$spacer * .5$
- **3** - (預設) 設定 **margin** 或是 **padding** 為 $\$spacer$
- **4** - (預設) 設定 **margin** 或是 **padding** 為 $\$spacer * 1.5$
- **5** - (預設) 設定 **margin** 或是 **padding** 為 $\$spacer * 3$
- **auto** - 設定 **margin** 為 **auto**