

JavaScript 簡介

邏輯分析與程式設計

國立雲林科技大學
王照明老師

助教：徐偉智

JavaScript簡介

- **JavaScript**原名LiveScript，於1995年在Netscape 2.0版正式發表，目前是市面上各大瀏覽器最普遍支援的直譯式腳本語言，其特色是**不需編譯**即可使用，撰寫和除錯十分容易。
- JavaScript大多數的功能皆是由系統所提供的物件來完成。所以可以使用的物件功能大多與**控制網頁**有關。
- JavaScript是讓HTML網頁可發揮**動畫**、**互動**與**動態特效**的程式語言。
- 使用JavaScript，可以設計出兼具**即時回饋**且有趣的**動態互動介面**。

JavaScript簡介

- JavaScript動態特效範例：
 - 範例1：<https://codepen.io/frank890417/pen/preQeE>
 - 範例2：<https://codepen.io/frank890417/pen/JpaRMK>
 - 範例3：<https://codepen.io/frank890417/pen/jZzqVY>
- JavaScript的執行結果可以在瀏覽器上以網頁的方式預覽。
- 因此，在正式開始學習JavaScript前，開發者必須先擁有網頁架構與網頁前端開發技術的基礎知識。

網頁前端架構簡介

- HTML + CSS + JavaScript
- 線上編輯器CodePen：<https://codepen.io/>

網頁前端開發技術

- 網頁前端開發一般指的是我們看到的網頁，包括裡面靜態的介面、一些酷炫的動態效果以及用戶的交互操作等。
- 基本的網頁前端開發技術包含：**HTML**、**CSS**與**JavaScript** (簡稱**JS**)，三者的關係如下圖：



HTML — 網頁基礎結構

- HTML網頁原始碼的基本結構分為數個區塊，分別標示網頁文件的不同用途。

- 基本結構標籤：

- `<html>` 標示是一份HTML網頁
- `<head>` HTML網頁的檔頭區塊
- `<title>` 網頁文件標題
- `<body>` HTML網頁的內容區塊

文件類型(DTD)宣告

```
<!doctype html>
```

網頁架構

```
<html>
```

檔頭資訊區

```
<head>
```

```
<meta charset="utf-8">
```

文件編碼宣告

```
<title>無標題文件</title>
```

網頁標題

```
</head>
```

網頁內容區

```
<body>
```

```
<p>這是一個段落</p>
```

段落內容

```

```

影像圖片

```
</body>
```

```
</html>
```

資料來源：http://tsweb44.com/TS_DWweb4/TSwrite/i7.html

HTML — 網頁基礎結構

- HTML語言是一種超文本標記語言，能夠使用標籤和屬性來呈現一些文本、圖片、表格、連結等內容，如下所示：

- **標籤(Tags)**：HTML標籤是一個字串符號，可以用來標示文字內容套用的編排格式，例如：在<p>開頭標籤和</p>結尾標籤之中的文字內容，就是使用預設格式編排成段落。

開頭標籤→

<p>這是一個測試網頁</p>

←結尾標籤(必須加斜線)

- **屬性(Attributes)**：每一個標籤可以擁有一些屬性來定義細部編排，例如：標籤的src、width和height屬性，可以指定顯示的圖形和尺寸的寬和高，如下所示：

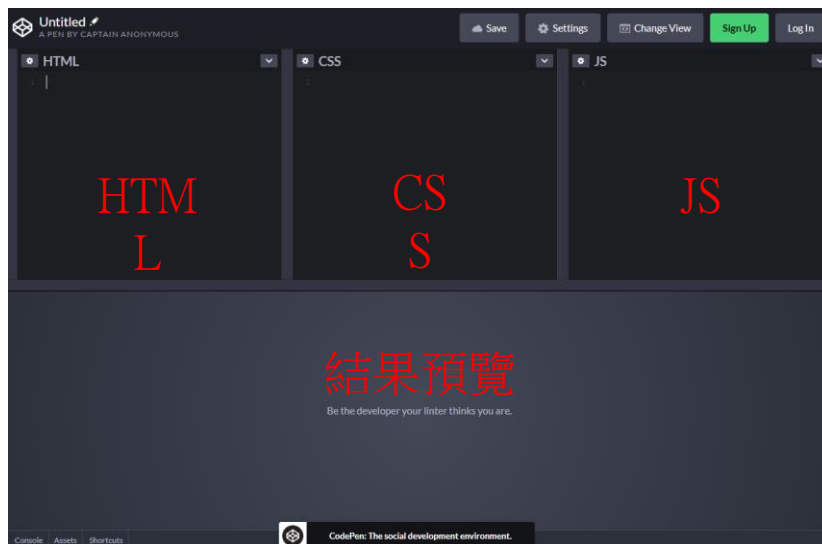
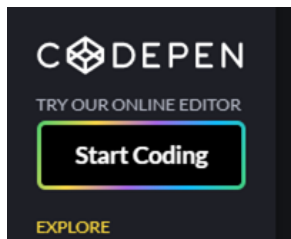
圖片路徑

圖片寬度

圖片高度

練習1 — HTML程式練習

- CodePen線上編輯器網址：<https://codepen.io/>
- 點選首頁左上角之「Start Coding」進入線上編輯畫面



練習1 — HTML程式練習

- 在HTML區塊建立一段文字段落 (使用<p>標籤)
- 新增一個按鈕 (使用<button>標籤)



The screenshot shows a dark-themed interface with a header bar containing a gear icon and the text "HTML". Below the header, there is a list of two code snippets, each with a dropdown arrow to its left:

```
1 ▾ <p>第一個程式練習</p>
2 ▾ <button>按我！</button>
```

- 於下方預覽區塊觀看程式結果



第一個程式練習

按我！

CSS — 美化網頁外觀

- 「CSS」（Cascading Style Sheets）層級式樣式表是一種樣式表語言，可以用來描述標示語言的顯示外觀和格式，能夠重新定義HTML元素的顯示效果。

↓ CSS規則名稱（選取器） 選取器可以是HTML標籤、元素名稱(id)或是類別(class)

```
body {  
  background-image: url(../images/background/a04.jpg);  
  background-attachment: fixed;  
  color: #FFFFFF;  
  margin: 0;  
}
```

↑ CSS屬性內容

CSS — 在HTML中嵌入

- 在HTML中嵌入CSS有三種方法：
 - 內嵌，放置在<style>和</style>標籤之間

```
<head>  
    < style type="text/css">  
        .....  
    </style>  
</head>
```

- 放置在有<link>標籤的href屬性指定的外部檔案中

```
<link rel="stylesheet" type="text/css" href="外部檔案路徑">
```

- 放在HTML的style屬性中

```
<button style="height:100px;">Button</button>
```

CSS — 美化網頁外觀

- CSS的目的是重新定義HTML元素的顯示樣式，例如：HTML標籤<p>是文字段落，預設使用瀏覽器的字體與字型大小，如果使用CSS，我們可以重新定義標籤<p>的顯示樣式，如下所示：

```
p {  
    font-size: 10pt;  
    color: red;  
}
```

- 程式碼定義<p>標籤使用尺寸10pt的文字，色彩為紅色，只要網頁使用<p>標籤的段落，都是套用此字型尺寸和色彩來顯示。

CSS — 美化網頁外觀

- **CSS規則名稱(選取器selector)**：選擇器，顧名思義就是選擇要改變樣式的元素，賦予其CSS的樣式規則。較單純的選取器種類分成以下幾種：
- **選擇全部 ***：「*」這個東西在選取器裡面扮演的就是全部事物，所以會選取到 HTML 內的所有元素。

```
* {  
    .....  
}
```
- **標籤選取器 Element**：通常是 HTML 的標籤名稱，被選取到的元素全部都會有相對應的樣式。

```
p {  
    .....  
}
```

CSS — 美化網頁外觀

- **ID選取器** `#IDName`：元素的id屬性其IDName與 `#IDName`相同就會被選取到。

HTML

```
<canvas id= “draw” ></canvas>
```

CSS

```
#draw {  
    .....  
}
```

- **類別選取器** `.ClassName`：class類似群組化管理的概念，元素的class屬性其ClassName與 `.ClassName`相同就會被選取到。

HTML

```
<canvas class= “group” ></canvas>
```

CSS

```
.group {  
    .....  
}
```

練習2 — CSS程式練習

HTML

```
1 <p>第一個程式練習</p>
2 <button id="click">按我！</button>
```

- 在HTML區塊賦予Button一個id值 (click)
- 在CSS區塊建立段落標籤<p>的樣式，設定顏色與大小
- 在CSS區塊建立按鈕(click)的樣式，設定長寬與顏色

CSS 每行程式的句尾 需加上分號「;」！

```
1 p{
2   font-size: 20pt;
3   color: #0000ff;
4 }
5 #click{
6   font-size: 20pt;
7   color: #ffffff;
8   background: #123456;
9   width: 150px;
10  height: 100px;
11 }
```

<p>標籤樣式：

將段落文字大小設為20pt
將段落文字顏色設為藍色(色碼)

id=click的按鈕樣式：

將按鈕文字大小設為20pt
將按鈕文字顏色設為白色(色碼)
將按鈕的背景顏色設為深藍色(色碼)
將按鈕的寬設為150pixel
將按鈕的高設為100pixel

第一個程式練習

按我！

JavaScript — 網頁特效與互動

- 透過JavaScript，我們可以修改網頁的各個方面，例如**內容**、**樣式**以及**與使用者互動的行為**。
- **網頁動態特效**通俗的說法便是通過JavaScript動態地控制CSS樣式(顏色、大小、位置、圖片更換…)，以達到實現動畫的效果。
- **與使用者互動的方式**包含按按鈕、滑鼠位置偵測、鍵盤控制、物理碰撞偵測等。
- 可以運用JavaScript在canvas畫布上**繪圖**。

JavaScript — 在HTML中嵌入

- 在HTML中嵌入JavaScript有三種方法：
 - 內嵌，放置在<script>和</script>標籤之間

```
<head>  
    <script type="text/javascript">  
        .....  
    </script>  
</head>
```

- 放置在有<script>標籤的src屬性指定的外部檔案中

```
<script type="text/javascript" src="外部檔案路徑"></script>
```

- 放在HTML事件處理程式中，例如：onclick或on

```
<button onclick="showjs()"></button>
```

JavaScript — 輸出訊息文字

- 在寫程式的過程中，經常會需要輸出一些訊息文字來進行除錯，在JavaScript中，最常使用的是`console.log()`，其餘輸出訊息文字的方法也有許多種：

- 一般訊息1 - `console.log("Hello World");` ← 最常使用
- 一般訊息2 - `console.info("一般訊息");`
- 除錯訊息 - `console.debug("除錯訊息");`
- 警告訊息 - `console.warn("警告訊息");`
- 錯誤提示 - `console.error("警告訊息");`
- 在網頁中顯示文字 - `document.write("網頁文字");`

JavaScript — 彈出視窗

- JavaScript可以實作網頁彈出視窗，語法如下：

```
alert(“Hello World!”);
```

訊息文字，輸出非數字的文字需加上雙引號



練習3 — JavaScript程式練習

- 為按鈕**加上onclick事件**，並填入函式名稱show()
- 在JS區塊撰寫function show()的程式，點擊按鈕，觸發彈出視窗與輸出訊息



HTML

```
1 <p>第一個程式練習</p>
2 <button id="click" onclick="show()">按我！</button>
```



JS

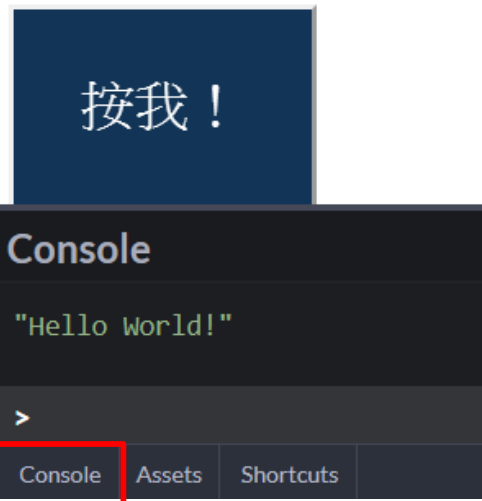
每行程式的句尾
需加上分號「;」！

```
1 function show(){
2     console.log("Hello World!");
3     alert("Hello World!");
4 }
```

練習3 — JavaScript程式練習

- Console.log()的結果會顯示在console視窗
- alert()的結果顯示在彈出視窗

第一個程式練習

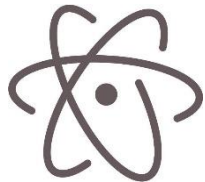


JavaScript開發環境建置

- Sublime Text3 程式編輯器
- Sublime Text官網：<https://www.sublimetext.com/>
- 於程式編輯器撰寫JavaScript程式

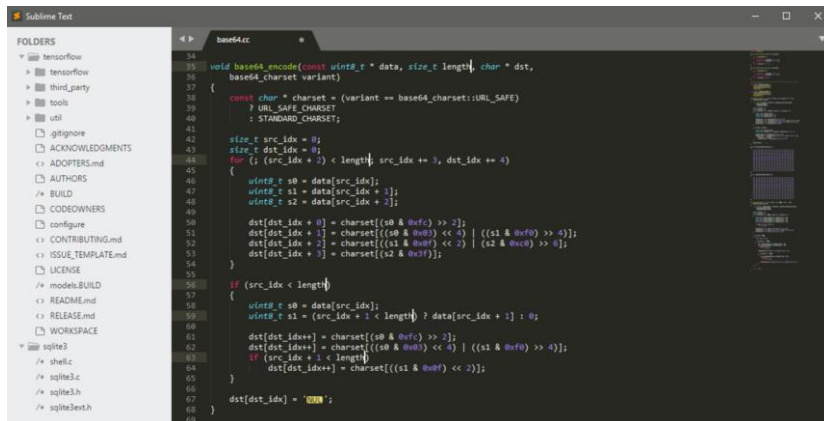
JavaScript要用什麼程式編輯器？

- 只要使用能編輯文字的桌面軟體就能寫JavaScript程式碼
- Atom、Sublime Text、Adobe Dreamweaver、NotePad++…甚至連記事本都可以寫JavaScript唷！



Sublime Text3 程式編輯器

- **Sublime Text**是一套跨平台的文字編輯器，可透過套件（Package）擴充功能。大多數的套件使用自由軟體授權釋出，並由社群建置維護。
- 官方網站：<https://www.sublimetext.com/>



```
34
35 void base64_encode(const uint8_t * data, size_t length, char * dst,
36                   base64_charset variant)
37 {
38     constexpr charset = (variant == base64_charset::URL_SAFE)
39         ? URL_SAFE_CHARSET
40         : STANDARD_CHARSET;
41
42     size_t src_idx = 0;
43     size_t dst_idx = 0;
44     for (; (src_idx + 2) < length; src_idx += 3, dst_idx += 4)
45     {
46         uint8_t s0 = data[src_idx];
47         uint8_t s1 = data[src_idx + 1];
48         uint8_t s2 = data[src_idx + 2];
49
50         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
51         dst[dst_idx + 1] = charset[(s0 & 0x03) << 4 | ((s1 & 0xf0) >> 4)];
52         dst[dst_idx + 2] = charset[(s1 & 0x0f) << 2 | (s2 & 0xc0) >> 6];
53         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
54     }
55
56     if (src_idx < length)
57     {
58         uint8_t s0 = data[src_idx];
59         uint8_t s1 = (src_idx + 1 < length) ? data[src_idx + 1] : 0;
60
61         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
62         dst[dst_idx++] = charset[(s0 & 0x03) << 4 | ((s1 & 0xf0) >> 4)];
63         if (src_idx + 1 < length)
64             dst[dst_idx++] = charset[(s1 & 0x0f) << 2];
65     }
66
67     dst[dst_idx] = '\0';
68 }
```

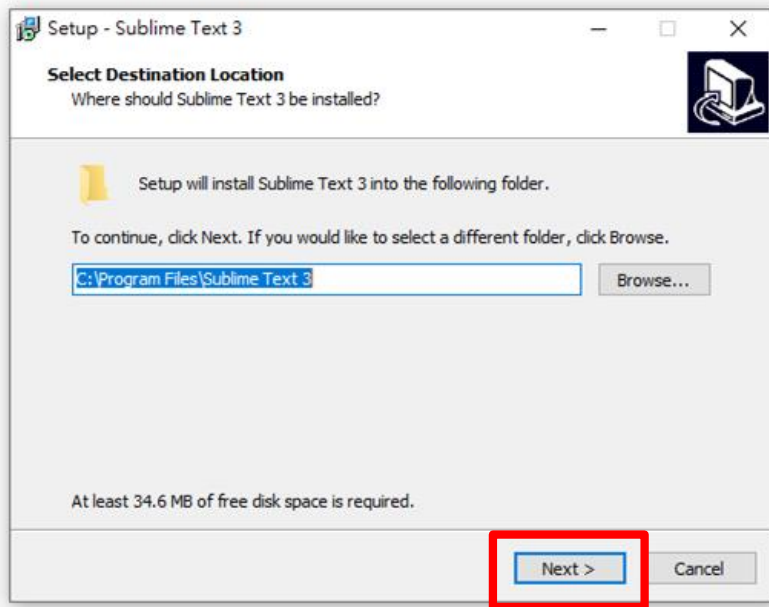

Sublime Text3 安裝教學

- 到Sublime Text官網下載安裝檔。
- 下載連結：<https://www.sublimetext.com/3>



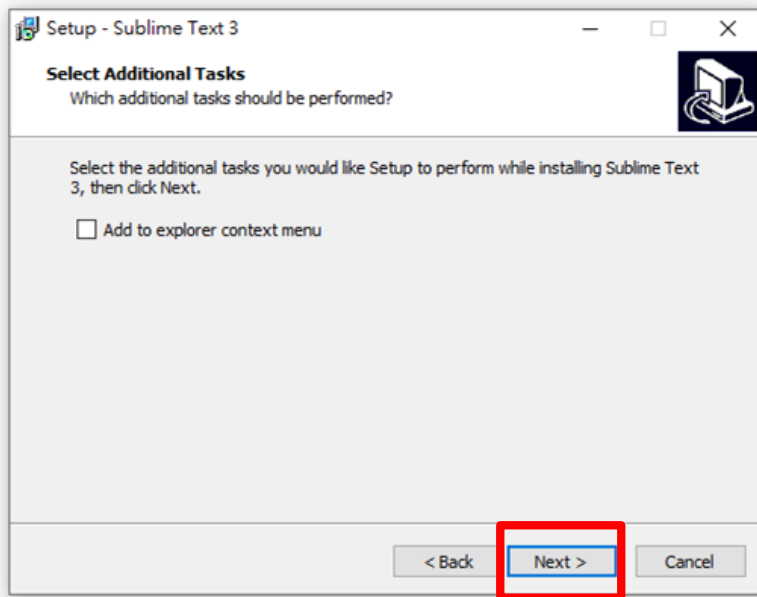
Sublime Text3 安裝教學

- 雙擊安裝檔，開始安裝Sublime Text，按「Next」。



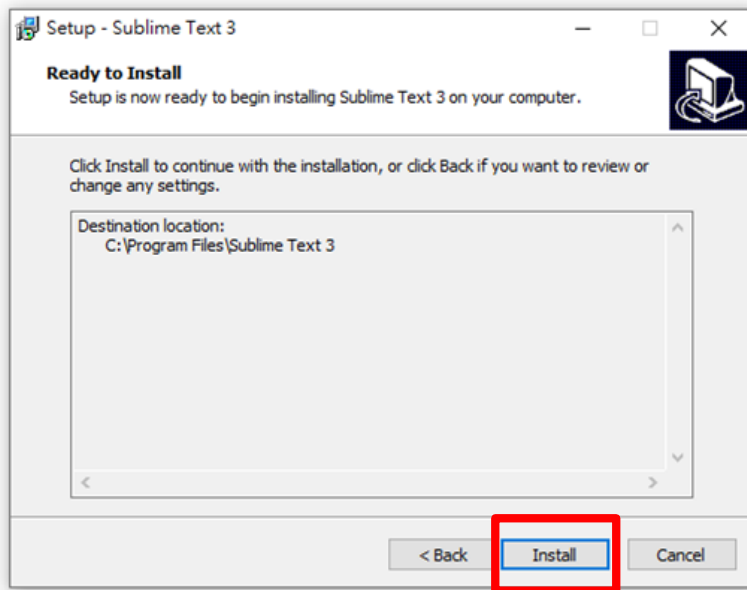
Sublime Text3 安裝教學

- 按「Next」進行下一步。



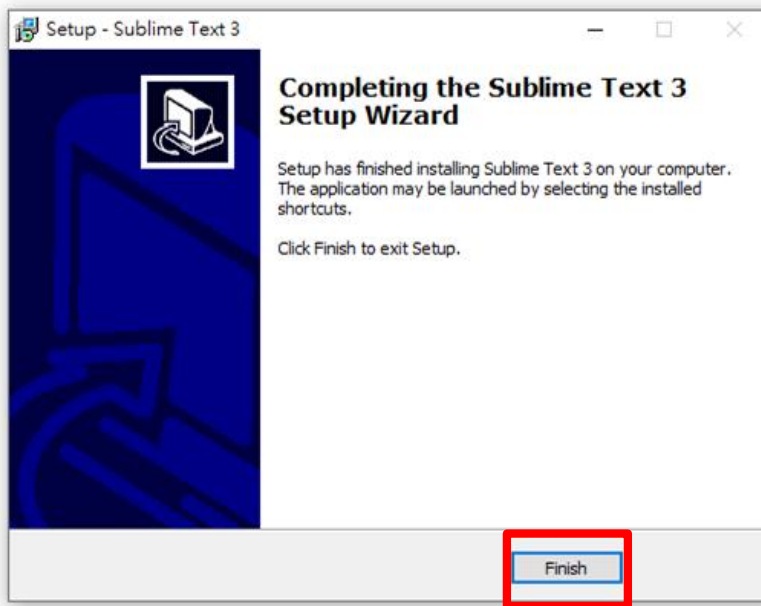
Sublime Text3 安裝教學

- 按「Install」開始安裝。



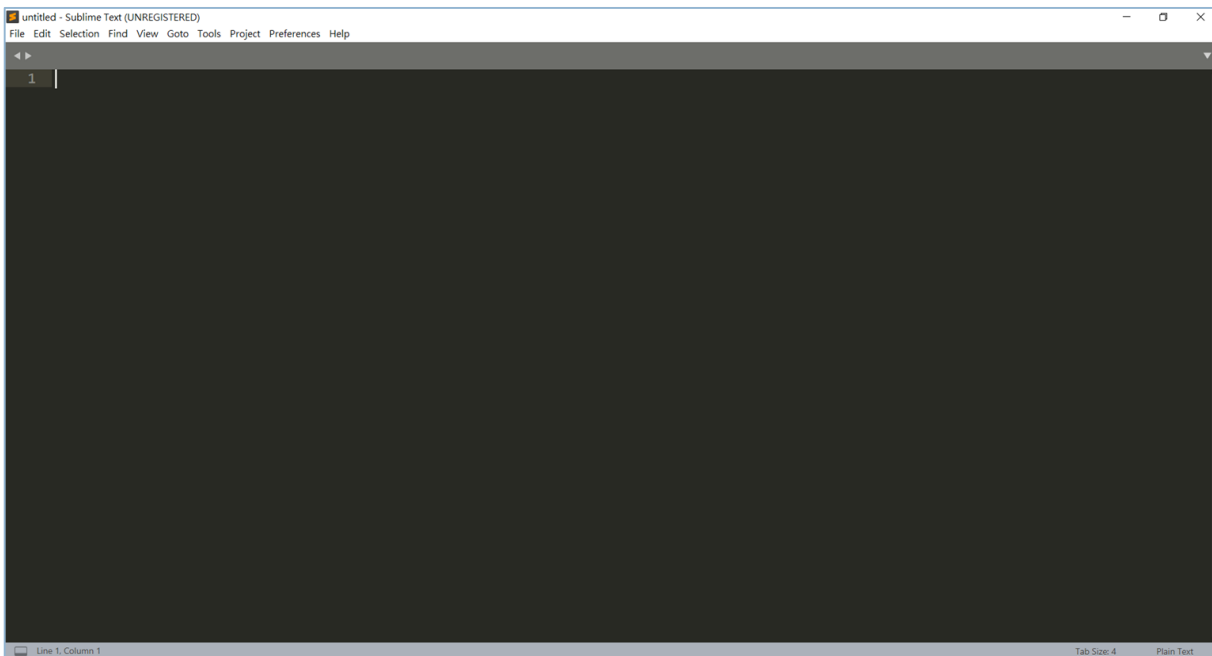
Sublime Text3 安裝教學

- 安裝完成，按「Finish」結束安裝程序。



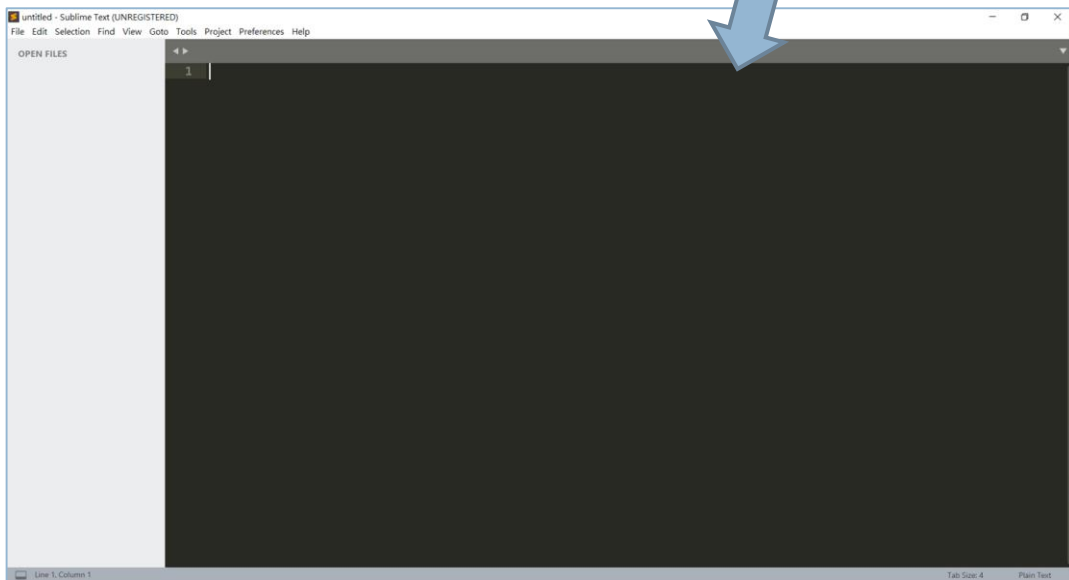
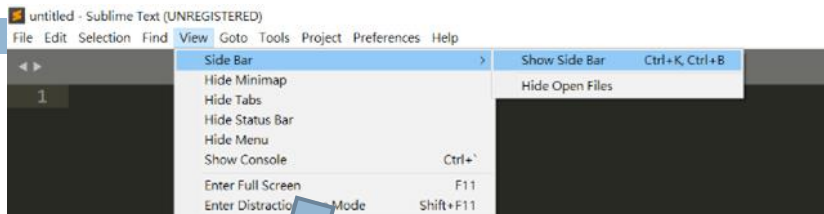
Sublime Text3 安裝教學

- 開啟Sublime，就可以開始使用了！



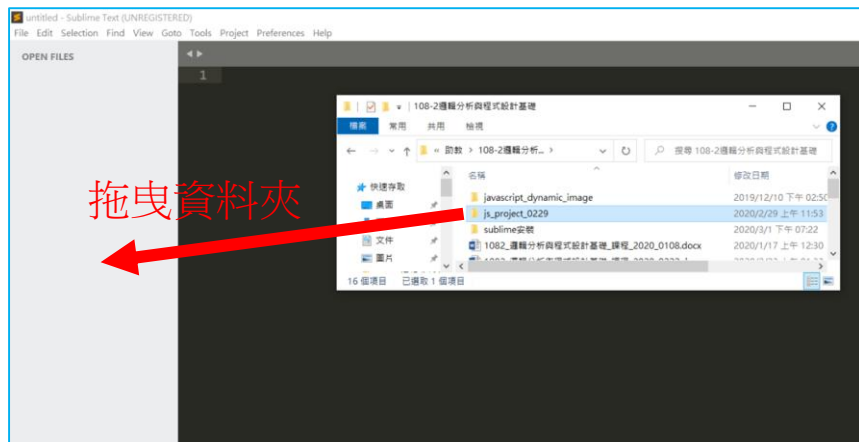
Sublime Text3 — 打開側邊Sidebar

- 開啟側邊SideBar



Sublime Text3 — 專案管理

- 將整個程式資料夾拖曳至側邊欄，便能在Sublime中管理專案，點擊檔案名稱就能編輯程式



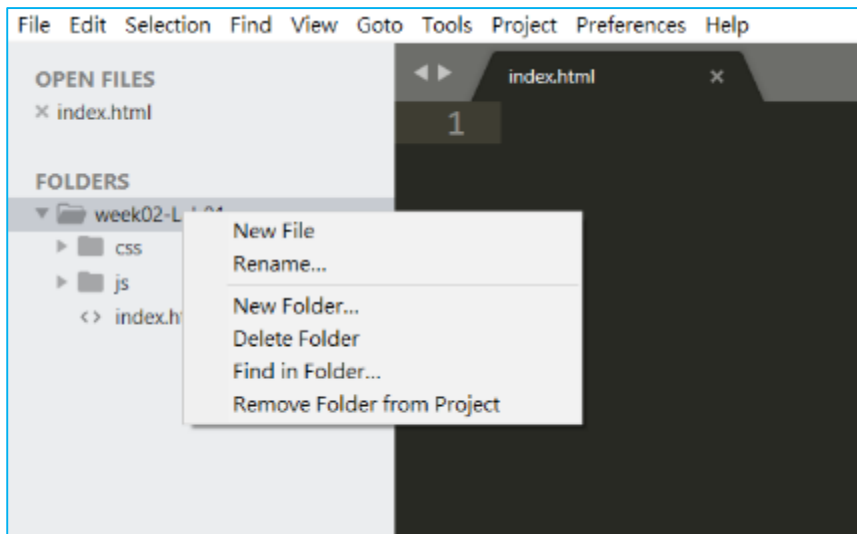
OPEN FILES

FOLDERS

- ▼ js_project_0229
 - ▶ js
 - <> 01_DrawBackground.html
 - <> 02_DrawImage.html
 - <> 03_ImageMove.html
 - <> 04_Boundaries.html
 - <> 05_ChangeColor.html
 - <> 06_finish.html
 - dvd-logo.png

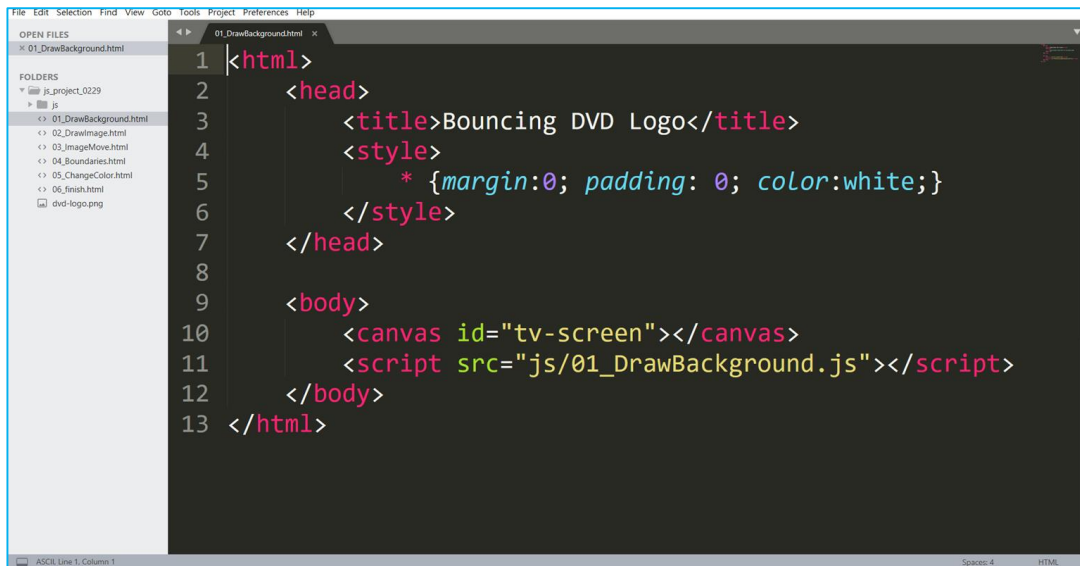
Sublime Text3 — 檔案管理

- 右鍵資料夾/檔案，可以做新增/修改/刪除等動作



Sublime Text3 — 放大/縮小文字

- 按住Ctrl不放，滑動滑鼠滾輪就能放大/縮小文字，可以調整至最舒適的大小



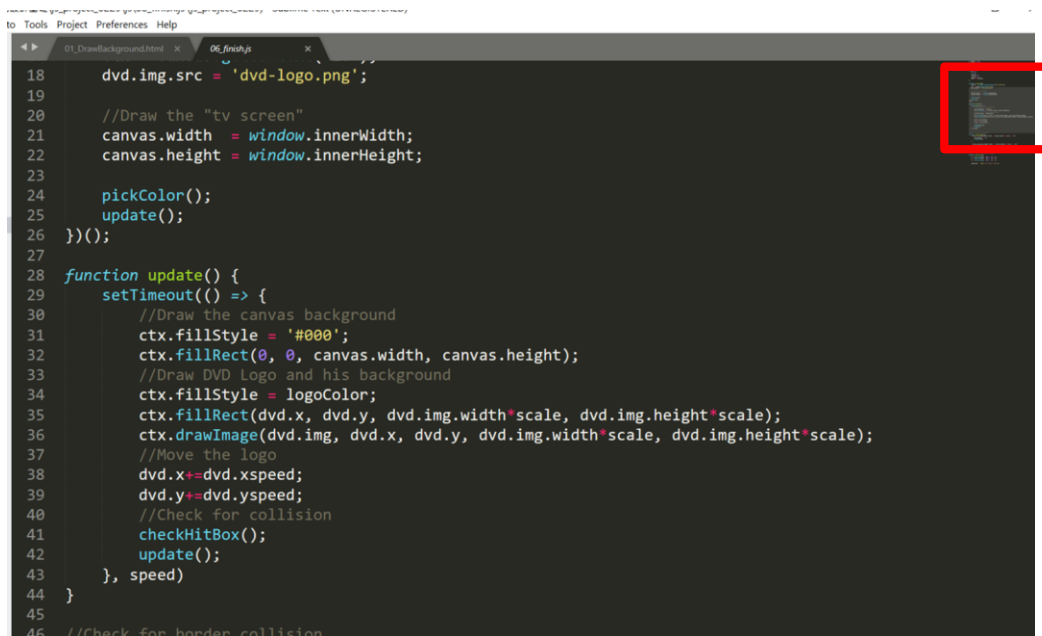
The screenshot shows the Sublime Text 3 interface with a dark theme. The left sidebar displays the 'FOLDERS' panel with a project named 'js_project_0229' containing several files. The main editor area shows an HTML file named '01_DrawBackground.html' with the following code:

```
1 <html>
2   <head>
3     <title>Bouncing DVD Logo</title>
4     <style>
5       * {margin:0; padding: 0; color:white;}
6     </style>
7   </head>
8
9   <body>
10    <canvas id="tv-screen"></canvas>
11    <script src="js/01_DrawBackground.js"></script>
12  </body>
13 </html>
```

The status bar at the bottom indicates 'ASCII Line 1, Column 1' on the left and 'Spaces: 4 HTML' on the right.

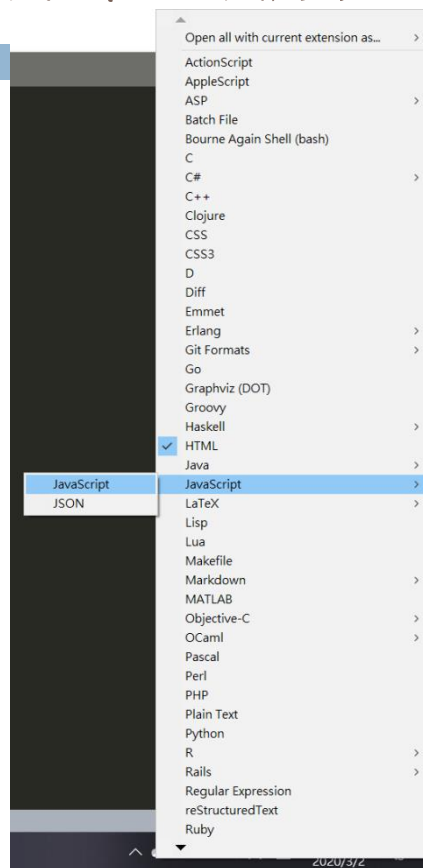
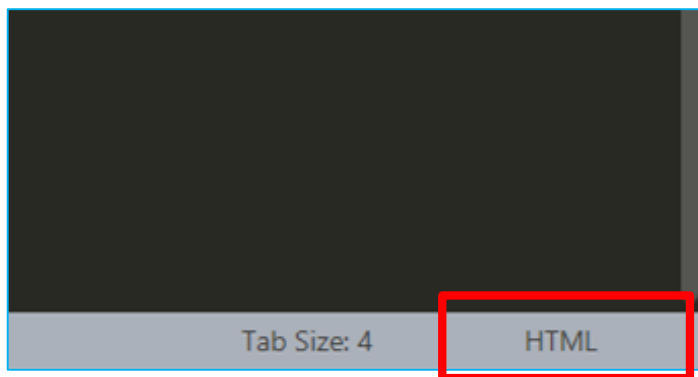
Sublime Text3 — 上下瀏覽

- 按住最右方的透明塊上下拖動，可以快速上下瀏覽程式



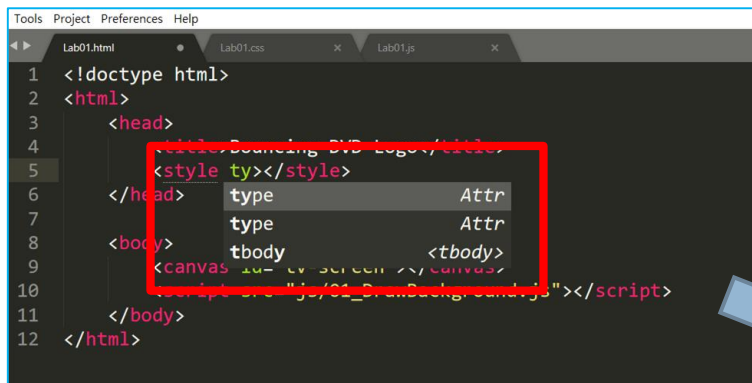
Sublime Text3 — 程式語言識別

- 開新檔案時，右下角可以依照需求切換程式語言

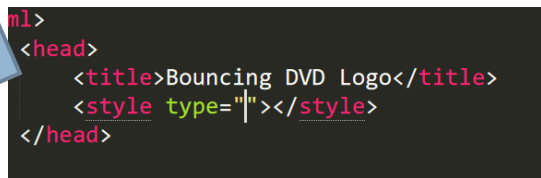


Sublime Text3 — 關鍵字

- 打出特定關鍵字，如雙引號 “、標籤結尾 </，Sublime 會自動補完後續的程式
- Sublime 可以根據所選程式語言，出現關鍵字
- 按tab鍵，可以選擇下拉選單的關鍵字，加快打字速度



```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Bouncing DVD Logo</title>
5     <style type=""></style>
6   </head>
7   <body>
8     <canvas id="cv" screen=""></canvas>
9     <script src="js/01_BackBackground.js"></script>
10
11 </body>
12 </html>
```



```
<html>
<head>
  <title>Bouncing DVD Logo</title>
  <style type=""></style>
</head>
```

Sublime Text3 常用快捷鍵一覽

一般操作(General)

快捷鍵	描述
Ctrl + Shift + P	打開命令框
Ctrl + K + B	開/關側邊欄

搜尋(Navigation/Goto)

快捷鍵	描述
Ctrl + F	打開底部的搜尋框
Ctrl + H	取代

Sublime Text3 常用快捷鍵一覽

Ctrl + /	註解單行
Ctrl + Shift + /	註解多行
Ctrl + Z	返回上一操作
Ctrl + Y	前往下一操作，對應 Ctrl + Z；或是重複最後執行的快捷鍵操作
Ctrl + Shift + V	貼上並進行縮排
Ctrl + Space	顯示自動化建議語句

更多快捷鍵請參考：<https://www.camdemy.com/media/6211/>

練習4 — 使用Sublime寫程式

- 練習程式下載連結：
https://drive.google.com/file/d/1wFsu2aSi1nzsgMpkol7R_roOys_4w2eI/view?usp=sharing
- 使用Sublime，將專案資料夾拖曳進左欄
- 複製練習3的完整程式碼(HTML、CSS、JS)，在Sublime中貼上 (想想看程式要貼在哪裡才正確?)

練習4 — 使用Sublime寫程式

```
Lab04.html x Lab04.css x Lab04.js x
1 <!doctype html>
2 <html>
3   <head>
4     <!-- 網頁標題文字 -->
5     <title>Week01 - Lab04 Practice</title>
6     <!-- CSS外部連結檔案 -->
7     <link rel="stylesheet" type="text/css" href="css/Lab04.css">
8   </head>
9   <body>
10    <!-- HTML程式放這裡 -->
11
12    <p>第一個程式練習</p>
13    <button id="click" onclick="show()">按我！</button>
14
15    <!-- JavaScript外部連結檔案 -->
16    <script src="js/Lab04.js"></script>
17  </body>
18 </html>
```

```
Lab04.html x Lab04.css x Lab04.js x
1 function show(){
2   console.log("Hello World!");
3   alert("Hello World!");
4 }
```

```
Lab04.html x Lab04.css x
1 p{
2   font-size: 20pt;
3   color: #0000ff;
4 }
5 #click{
6   font-size: 20pt;
7   color: #ffffff;
8   background: #123456;
9   width: 150px;
10  height: 100px;
11 }
```

練習4 — 在瀏覽器預覽結果

- 在資料夾中找到HTML檔案，雙擊用瀏覽器開啟(以Google Chrome為例)



練習4 — 在瀏覽器預覽結果

- 按按鈕，看有沒有反應

這個網頁顯示

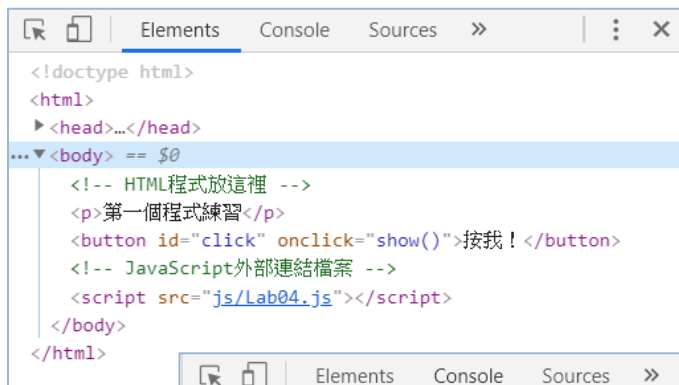
Hello World!

確定

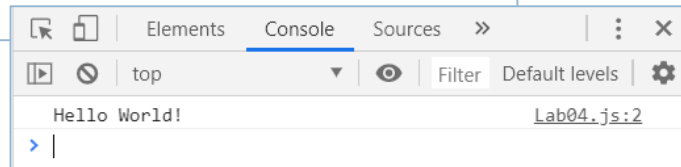
- 按F12，開啟Chrome開發者工具，可以看到：

- 網頁原始碼
- CSS
- Console結果
- ……等

- 如果HTML檔案有新變動，按F5
可以重新整理網頁



```
<!doctype html>
<html>
  <head>...</head>
  <body> == $0
    <!-- HTML程式放這裡 -->
    <p>第一個程式練習</p>
    <button id="click" onclick="show()">按我！</button>
    <!-- JavaScript外部連結檔案 -->
    <script src="js/Lab04.js"></script>
  </body>
</html>
```



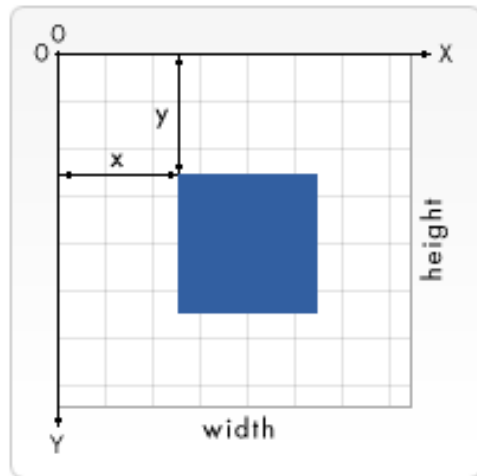
```
top
Hello World! Lab04.js:2
> |
```

Canvas畫布與JavaScript

Canvas畫布

- 最新版的HTML5語言新增了**Canvas畫布標籤**，開發者可以使用JavaScript語言在上面繪圖、合成圖照片、建立動畫、甚至處理即時的影片播放。
- canvas 本身帶有基本的寬高屬性，可以直接在元素上調整畫布的大小；另外也支援我們使用 CSS 來調整畫布的一些基本樣式。

網格的原點 (座標 $(0, 0)$) 坐落於左上角，所有元素定位皆相對於此左上角原點



```
<canvas id="myCanvas" width="300" height="300"></canvas>
```

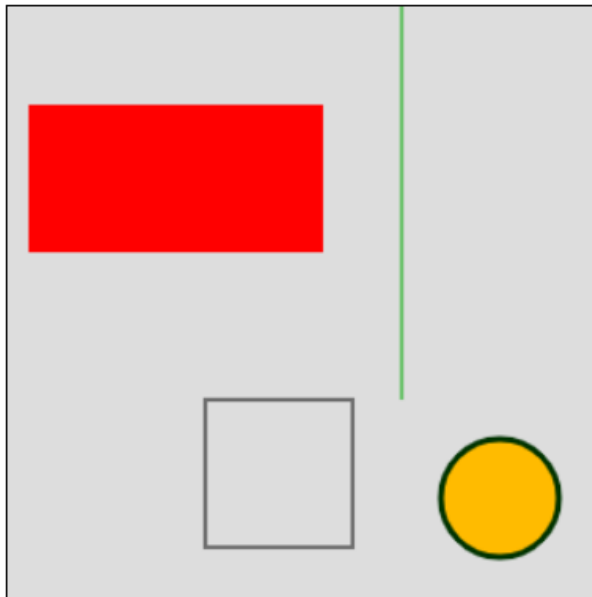
元素名稱

畫布寬度

畫布高度

練習5 — 繪製幾何圖形

- 在畫布中繪製矩形、圓形與直線。



練習5 — 繪製幾何圖形

□ HTML程式

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Week02 - Lab05 Practice</title>
5     <link rel="stylesheet" type="text/css" href="css/Lab05.css">
6   </head>
7   <body>
8     <canvas id="canvas-draw" width="300" height="300"></canvas>
9     <script src="js/Lab05.js"></script>
10  </body>
11 </html>|
```

練習5 — 繪製幾何圖形

□ CSS程式

```
1  p{
2      font-size: 20px;
3      color: #0000ff;
4  }
5  canvas{ //將canvas置於網頁中央
6      padding: 0;
7      margin: auto;
8      display: block;
9  }
10 #canvas-draw{
11     background: #dddddd;
12     border: 1px #000000 solid;
13 }
```


練習5 — 繪製幾何圖形

- Canvas可視為是一個物件，在JavaScript中，我們首先必須先獲取canvas畫布的上下文物件(context)，才能進一步在畫布上繪製圖形。

```
var canvas = document.getElementById("draw");
```

創建canvas變數，存取id為draw的canvas物件

```
var draw2d = canvas.getContext("2d");
```

創建draw2d變數，獲取canvas畫布的上下文物件，方便調用相關函數做使用 (往後的課程會詳細介紹)

練習5 — 繪製幾何圖形

- 繪製矩形程式：

```
draw2d.fillStyle = "#ff0000"; //矩形色碼
```

```
draw2d.fillRect(10, 50, 150, 75); _____ //繪製矩形(填滿)
```

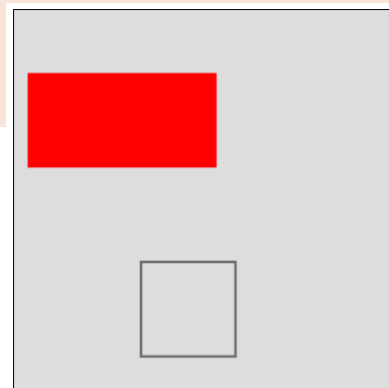
X座標, Y座標, 寬度, 高度

```
draw2d.strokeRect(100, 200, 75, 75); //繪製矩形路徑(只有邊框)
```

X座標, Y座標, 寬度, 高度

- 矩形的繪製以左上角為起點(x座標、y座標)，寬度與長度從起點

繼續向右下延伸成矩形



練習5 — 繪製幾何圖形

- 繪製線條程式：

```
draw2d.strokeStyle = "#00aa00"; //線條色碼
```

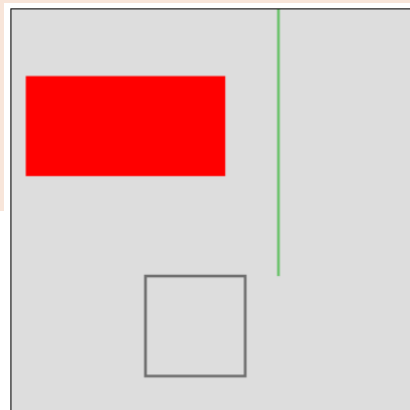
```
draw2d.moveTo(200, 0); //移到這個座標點(起始點)
```

X座標, Y座標

```
draw2d.lineTo(200, 200); //線條結束點
```

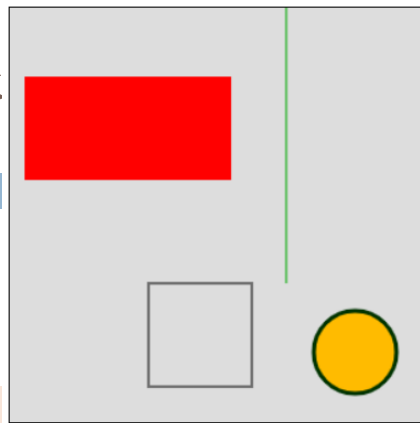
X座標, Y座標

```
draw2d.stroke(); //畫出線條路徑
```



練習5 — 繪製幾何圖形

□ 繪製圓形程式：



```
draw2d.beginPath(); //產生一個新路徑
```

```
draw2d.arc(250, 250, 30, 0, 2 * Math.PI, false); //建立圓形
```

中心點X座標,Y座標,半徑,起始角,結束角,順/逆時針(順=false 逆=true)

```
draw2d.fillStyle = "#ffbb00"; //圓形顏色
```

```
draw2d.fill(); //填滿(圓形)
```

```
draw2d.lineWidth = 3; //線條寬度
```

```
draw2d.strokeStyle = "#003300"; //線條顏色
```

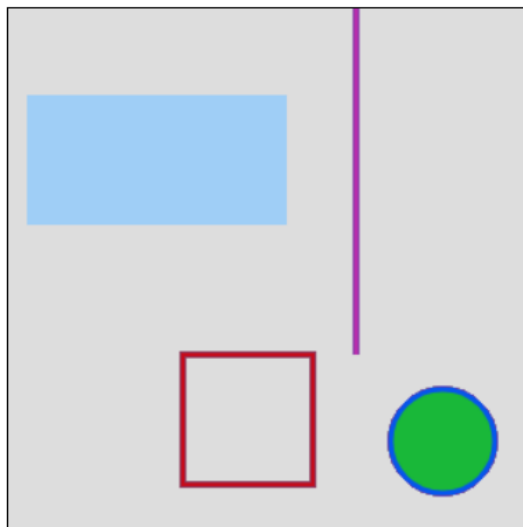
```
draw2d.stroke(); //繪製路徑
```

□ JS程式

```
1
2  var canvas = document.getElementById("canvas-draw");
3  var draw2d = canvas.getContext("2d");
4
5  //繪製矩形
6  draw2d.fillStyle = "#ff0000";
7  draw2d.fillRect(10, 50, 150, 75);
8
9  //繪製矩形邊框
10 draw2d.strokeRect(100, 200, 75, 75);
11
12 //繪製線條
13 draw2d.strokeStyle = "#00aa00";
14 draw2d.moveTo(200, 0);
15 draw2d.lineTo(200, 200);
16 draw2d.stroke();
17
18 //繪製圓形
19 draw2d.beginPath();
20 draw2d.arc(250, 250, 30, 0, 2 * Math.PI, false);
21 draw2d.fillStyle = "#ffbb00";
22 draw2d.fill();
23
24 //繪製圓形邊框
25 draw2d.lineWidth = 3;
26 draw2d.strokeStyle = "#003300";
27 draw2d.stroke();
```

練習6 — 隨機變換顏色

- 在畫布中繪製矩形、圓形與直線，按下按鈕要能隨機變換顏色。



換顏色

練習6 — 隨機變換顏色

□ HTML程式

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Week02 - Lab06 Practice</title>
5     <link rel="stylesheet" type="text/css" href="css/Lab06.css">
6   </head>
7   <body>
8     <canvas id="canvas-draw" width="300" height="300"></canvas>
9     <div>
10      <button id="button-color" onclick="changeColor()">換顏色</button>
11    </div>
12    <script src="js/Lab06.js"></script>
13  </body>
14 </html>
```

新增這段

□ CSS程式

```
1  p{
2    font-size: 20px;
3    color: #0000ff;
4  }
5  canvas{
6    padding: 0;
7    margin: auto;
8    display: block;
9  }
10 div{ //讓按鈕能夠置中顯示
11     text-align: center;
12     margin-top: 20px;
13 }
14 #canvas-draw{
15     background: #dddddd;
16     border: 1px #000000 solid;
17 }
18 #button-color{ //按鈕外觀
19     width: 300px;
20     height: 30px;
21     background: #123456;
22     color: #ffffff;
23     border: none;
24 }
```

新增這兩段

□ JS程式

```
1  var canvas = document.getElementById("canvas-draw");
2  var draw2d = canvas.getContext("2d");
3
4  drawShape(); 將繪圖程式放進名為drawShape()的function
5
6  function drawShape(){ 將色碼替換成randomColor()
7      //繪製矩形
8      draw2d.fillStyle = randomColor();
9      draw2d.fillRect(10, 50, 150, 75);
10     draw2d.strokeRect(100, 200, 75, 75);
11
12     //繪製線條
13     draw2d.strokeStyle = randomColor();
14     draw2d.moveTo(200, 0);
15     draw2d.lineTo(200, 200);
16     draw2d.stroke();
17
18     //繪製圓形
19     draw2d.beginPath();
20     draw2d.arc(250, 250, 30, 0, 2 * Math.PI, false);
21     draw2d.fillStyle = randomColor();
22     draw2d.fill();
23
24     //繪製圓形外框
25     draw2d.lineWidth = 3;
26     draw2d.strokeStyle = randomColor();
27     draw2d.stroke();
28 }
```

練習6 — 隨機變換顏色

新增這兩個

function

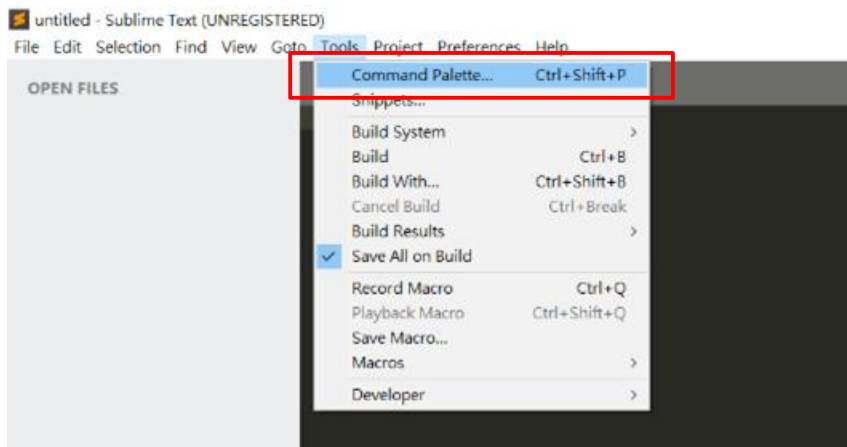
```
30 function changeColor(){//變換顏色時需重繪圖形
31     drawShape();
32 }
33
34 function randomColor(){ //RGB亂數程式
35     var r = Math.random() * (256 - 0) + 0;
36     var g = Math.random() * (256 - 0) + 0;
37     var b = Math.random() * (256 - 0) + 0;
38     var newColor = 'rgb('+r+', '+g+', '+b+')';
39     return newColor;
40 }
```

□ JS程式

Sublime Text 相關套件安裝

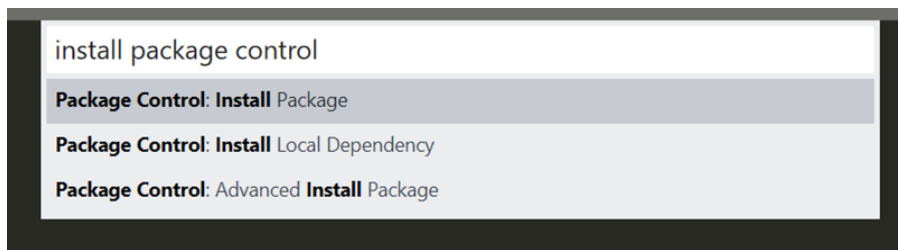
Sublime套件 — Package Control

- Package Control是一個可以管理Sublime所有套件的強大套件，可以輕鬆安裝/解除安裝套件
- Tool > Command Palette 開啟指令面板
(Win/Linux: [ctrl+shift+p](#), Mac: [cmd+shift+p](#))



Sublime套件 — Package Control

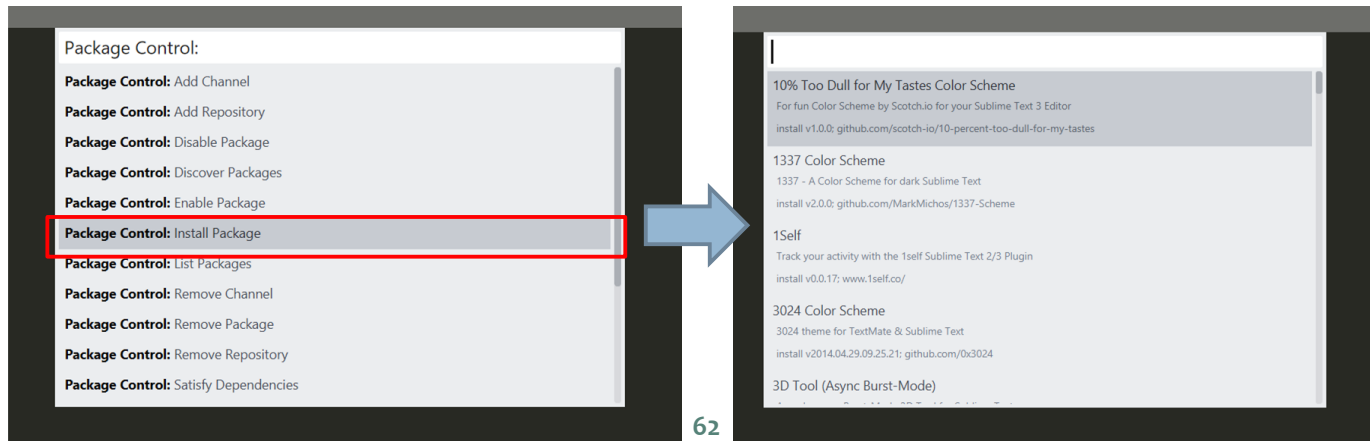
- 輸入「Install Package Control」並按Enter
- 等待一下即安裝完成



- 安裝好Package Control後，我們再來安裝一些能讓我們更方便開發JavaScript的套件吧！

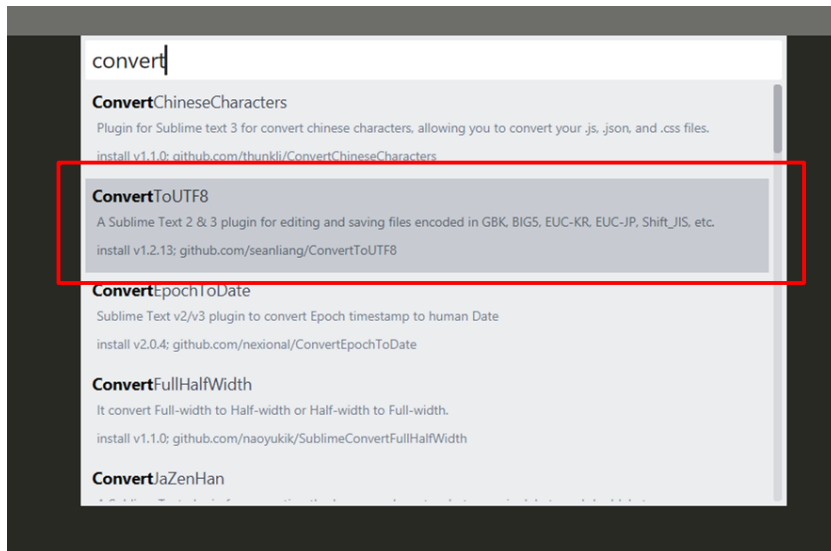
Package Control安裝套件方法

- Preference > Package Control 開啟指令面板
- 選擇Package Control: Install Package
- 進入套件搜尋畫面
- 使用「ctrl+`」(ctrl+數字鍵1左邊那顆)能開啟console，查看安裝進度



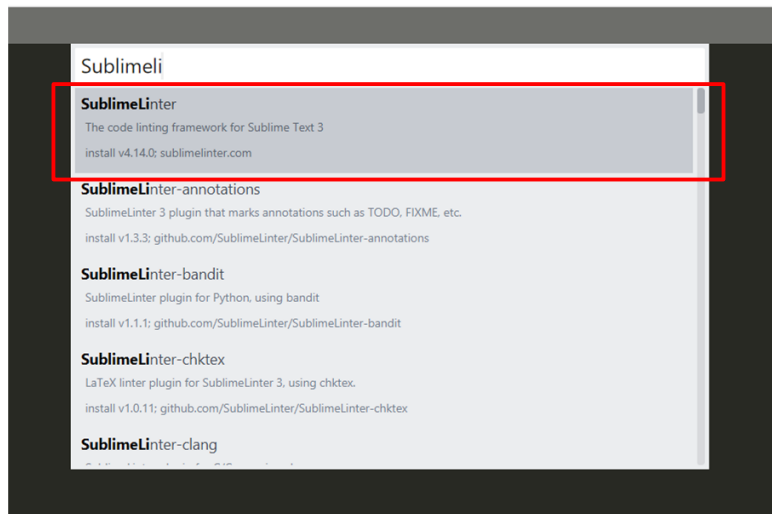
Sublime編碼套件 — ConvertToUTF8

- 在套件搜尋畫面搜尋「ConvertToUTF8」
- 此套件可以確保不同編碼文件打開不會出現亂碼



Sublime偵錯套件 — SublimeLinter

- 在套件搜尋畫面搜尋「SublimeLinter」
- 此套件可以幫助檢查程式語法是否有錯誤，需根據想偵錯的語言下載附加元件



Sublime偵錯套件 — SublimeLinter

- 安裝Node.js，輔助安裝SublimeLinter的偵錯功能
- Node.js下載：<https://nodejs.org/en/>



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)

下載較穩定
版本並安裝

12.16.1 LTS

Recommended For Most Users

13.9.0 Current

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

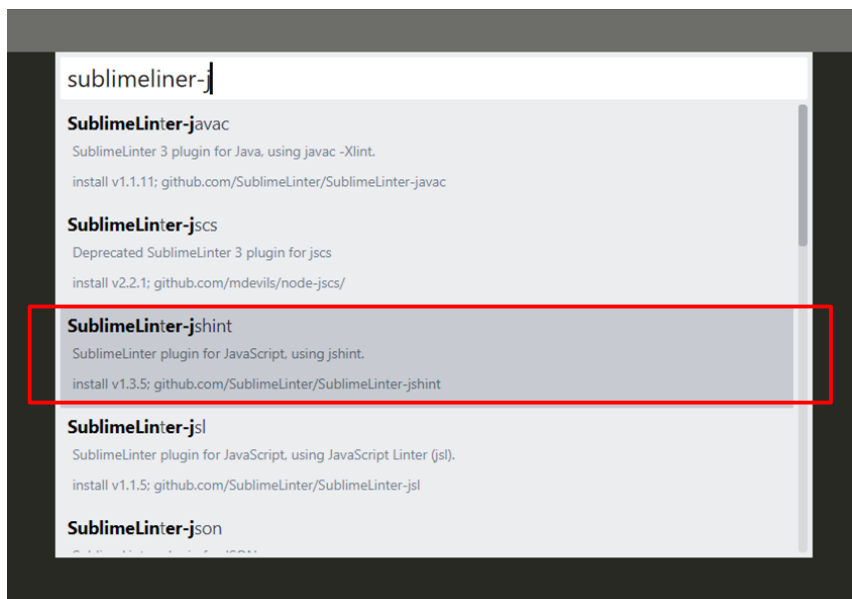
[Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.

Sublime偵錯套件 — SublimeLinter

- 要讓系統檢查JavaScript的語法，可以安裝相關套件輔助
- Install Package>搜尋「[SublimeLinter-jshint](#)」附加元件並安裝



Sublime偵錯套件 — SublimeLinter

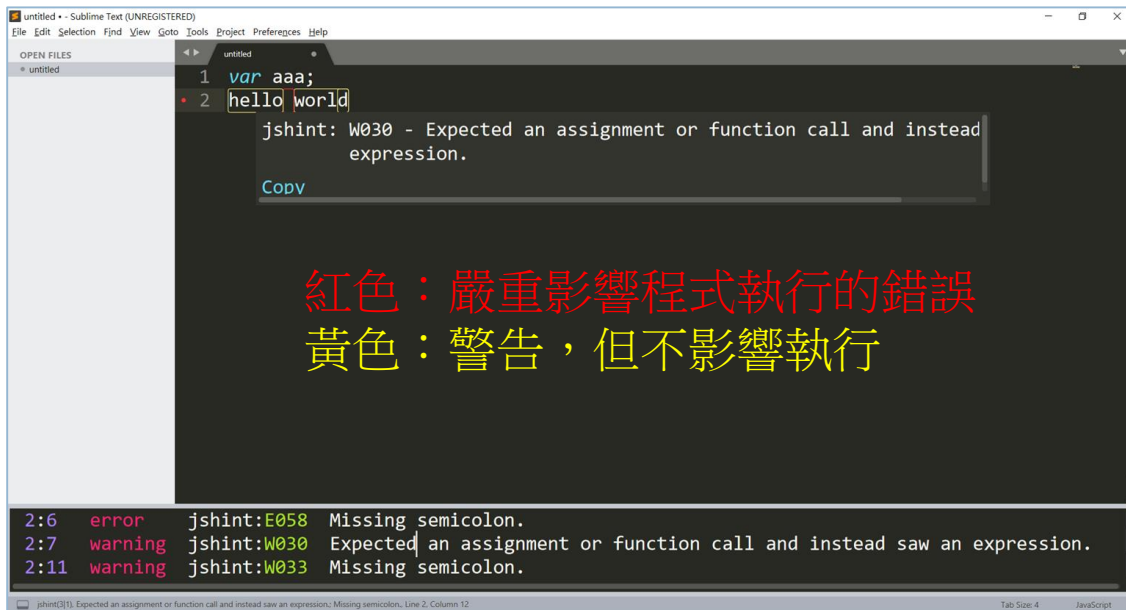
- 到命令提示字元，輸入「`npm install -g jshint`」
- 安裝成功後重開Sublime，就能開啟JS偵錯功能



```
命令提示字元
Microsoft Windows [版本 10.0.18363.657]
(c) 2019 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\User>npm install -g jshint
C:\Users\User\AppData\Roaming\npm\jshint -> C:\Users\User\AppData\Roaming\npm\node_modules\jshint\bin\jshint
+ jshint@2.11.0
added 31 packages from 15 contributors in 3.052s
C:\Users\User>
```

Sublime偵錯套件 — SublimeLinter

- 輸入錯誤的程式碼，SublimeLinter會即時偵錯，幫助程式開發者找Bug



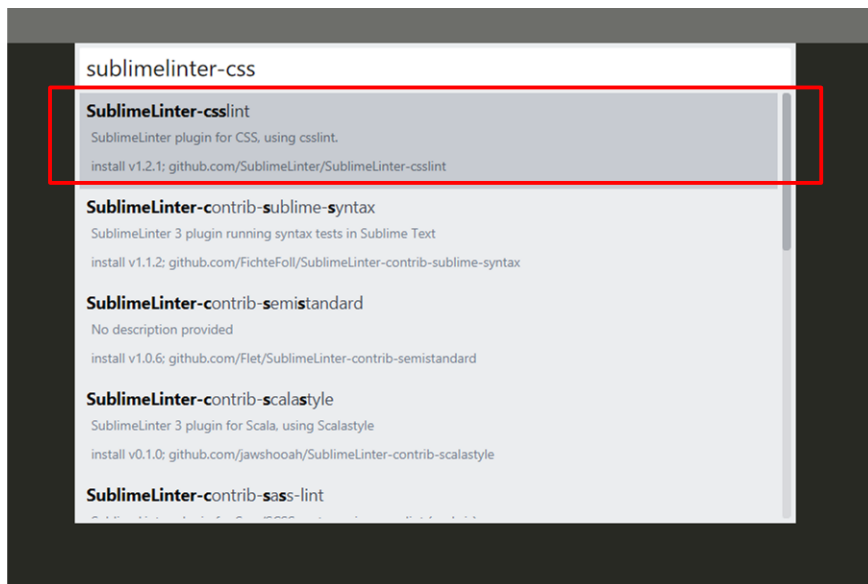
The screenshot shows the Sublime Text editor interface. The main editor area displays two lines of JavaScript code: `1 var aaa;` and `2 hello world`. A tooltip for the second line shows the error: `jshint: W030 - Expected an assignment or function call and instead expression.` with a 'Copy' button. The bottom status bar shows three error messages: `2:6 error jshint:E058 Missing semicolon.`, `2:7 warning jshint:W030 Expected an assignment or function call and instead saw an expression.`, and `2:11 warning jshint:W033 Missing semicolon.`

```
1 var aaa;  
2 hello world  
   jshint: W030 - Expected an assignment or function call and instead  
   expression.  
   Copy
```

2:6 error jshint:E058 Missing semicolon.
2:7 warning jshint:W030 Expected an assignment or function call and instead saw an expression.
2:11 warning jshint:W033 Missing semicolon.

Sublime偵錯套件 — SublimeLinter

- CSS與HTML偵錯套件的安裝方式與JavaScript差不多
- Install Package>安裝「[SublimeLinter-csslint](#)」附加元件



Sublime偵錯套件 — SublimeLinter

- 到命令提示字元，輸入「`npm install -g csslint`」
- 安裝成功後重開Sublime，就能開啟CSS偵錯功能



```
命令提示字元
Microsoft Windows [版本 10.0.18363.657]
(c) 2019 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\User>npm install -g csslint
C:\Users\User\AppData\Roaming\npm\csslint -> C:\Users\User\AppData\Roaming\npm\node_modules\csslint\dist\cli.js
+ csslint@1.0.5
added 3 packages from 25 contributors in 2.469s
C:\Users\User>
```

Sublime偵錯套件 — SublimeLinter

- 接下來繼續安裝HTML的偵錯套件
- Install Package>安裝「[SublimeLinter-contrib-htmlhint](#)」附加元件



Sublime偵錯套件 — SublimeLinter

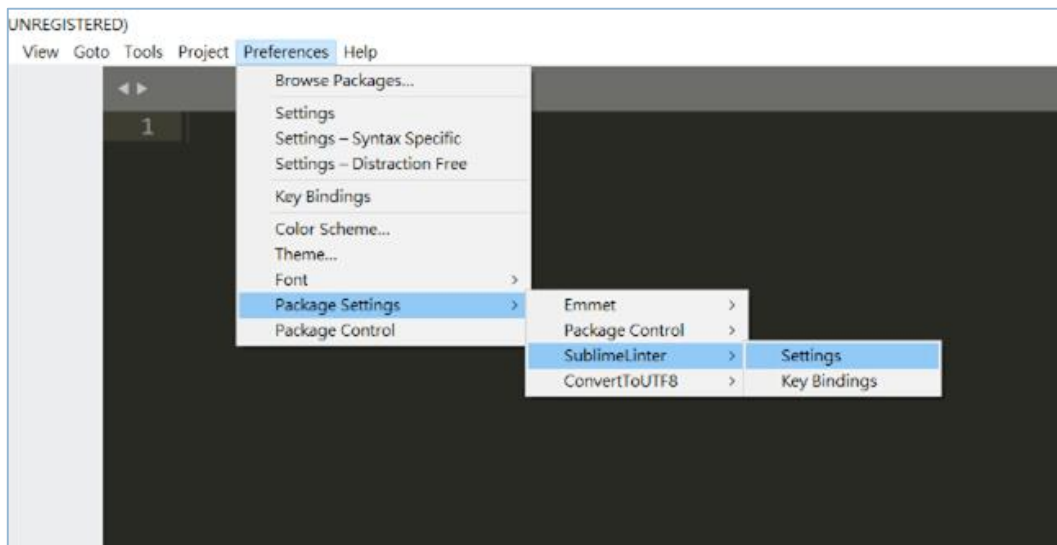
- 到命令提示字元，輸入「`npm install -g htmlhint`」
- 若出現版本相容錯誤也沒關係，偵錯套件依然能正常運行
- 安裝成功後重開Sublime，就能開啟HTML偵錯功能

選取 命令提示字元

```
Microsoft Windows [版本 10.0.18363.657]  
(c) 2019 Microsoft Corporation. 著作權所有，並保留一切權利。  
  
C:\Users\User>npm install -g htmlhint  
npm WARN deprecated request@2.88.0: request has been deprecated, see https://github.com/request/request/issues/3142  
C:\Users\User\AppData\Roaming\npm\htmlhint -> C:\Users\User\AppData\Roaming\npm\node_modules\htmlhint\bin\htmlhint  
+ htmlhint@0.11.0  
added 94 packages from 112 contributors in 5.539s
```


Sublime偵錯套件 — SublimeLinter

- SublimeLinter預設隨時都在背景執行偵錯功能
- Preference > SublimeLinter > Setting 可以開啟設定腳本，改變偵錯模式



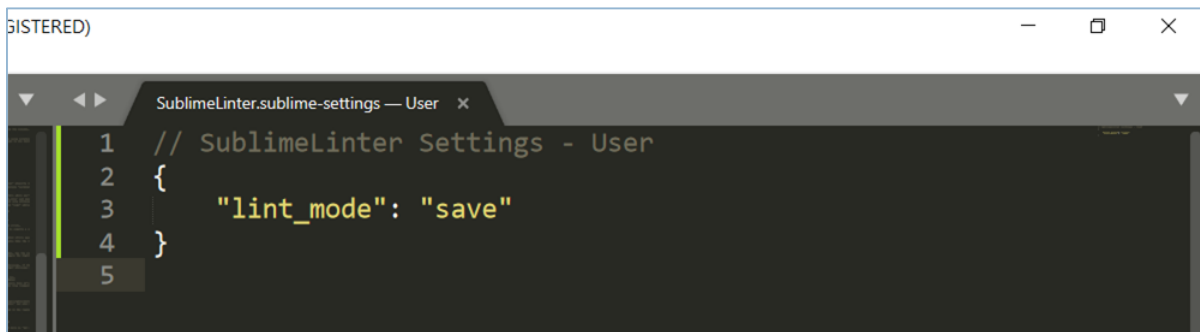
Sublime偵錯套件 — SublimeLinter

- 找到左欄程式的第51-56行，便能看到偵錯模式的設定分為四種：
 - **Background**：每當檔案發生變更便立即偵錯 (不推薦)
 - **Load / Save**：載入檔案 / 儲存檔案時偵錯
 - **Manual**：平時不偵錯，除非手動開啟偵錯功能
 - **Save**：儲存檔案時偵錯

```
50
51 // Lint Mode determines when the linter is run.
52 // - background: asynchronously on every change
53 // - load_save: when a file is opened and every time it's
54 // - manual: only when calling the Lint This View command
55 // - save: only when a file is saved
56 "lint_mode": "background",
57
```

Sublime偵錯套件 — SublimeLinter

- 選定適合自己的偵錯模式
- 假設選擇Background以外的選項，在右方欄第三行輸入：
 - 開檔/儲存時偵錯 - **"lint_mode": "load_save"**
 - 手動偵錯 - **"lint_mode": "manual"**
 - 儲存時偵錯 - **"lint_mode": "save"**



```
1 // SublimeLinter Settings - User
2 {
3     "lint_mode": "save"
4 }
5
```