



# 設定views



designed by freepik

Estimated time:

50 min.

III 資訊工業策進會 Institute for Information Industry

【Key Points】：

## 學習目標

- 12-1: 安裝樣版引擎 EJS
- 12-2: 設定 EJS
- 12-3: EJS Tags



12-1

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

我們即將一起學習：

- 安裝 ejs 樣板引擎
- 透過yarn安裝樣板引擎
- express-generator專案產生器
- 以 git clone 的方式安裝 ejs
- 什麼是樣板引擎
- EJS樣板引擎
- EJS 設定
- 如何傳遞參數到頁面樣板
- 常用EJS標籤
- 基本頁面元素
- 流程控制標籤

【Key Points】：

12-1: 安裝樣版引擎 EJS

12-2: 設定 EJS

12-3: EJS Tags

## 12-1: 安裝樣版引擎 EJS

- 安裝 ejs 樣板引擎
- 透過 yarn 安裝樣板引擎
- express-generator 專案產生器
- 以 git clone 的方式安裝 ejs



designed by freepik



designed by freepik

12-2

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

- 輸入 npm install ejs 指令，安裝 ejs
- 輸入以下指令檢查安裝版本: npm ejs -v
- 透過yarn安裝樣板引擎，指令: yarn add ejs
- express-generator是專案產生器，它可以自動幫你產生一個 express 專案，且透過參數就能安裝 view engine，指令: express -e {專案名稱}
- 安裝套件的方式，除了透過套件管理工具，還可以指定github網址

### 【Key Points】：

輸入 npm install ejs 指令，安裝 ejs

透過yarn安裝樣板引擎

安裝套件的方式，除了透過套件管理工具，還可以指定github網址

# 安裝 ejs 樣板引擎

- **如何安裝EJS樣板引擎:**

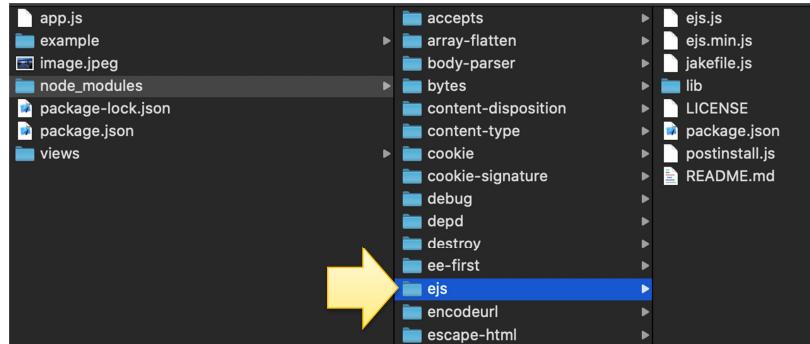
1. 在 Visual Studio Code 按下「Ctrl + 滑鼠」開啟 terminal 終端機視窗
2. 輸入: `npm install ejs`

- **如何確認安裝成功**

- 輸入以下指令

`npm ejs -v`

- 手動檢查 ejs 安裝，  
打開資料夾  
`node_modules`，  
如果發現 ejs 資料夾，  
代表安裝成功



12-3

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

1. 在 Visual Studio Code 按下「Ctrl + 滑鼠」開啟 terminal 終端機視窗
2. 輸入 `npm install ejs` 指令，安裝 ejs
3. 輸入以下指令檢查安裝版本: `npm ejs -v`
4. 如安裝成功，會看見版本號
5. 或手動檢查 ejs 安裝，可以打開資料夾 `node_modules`，如果發現 ejs 資料夾，代表安裝成功

## 【Key Points】：

安裝: `npm install ejs`

檢查安裝版本: `npm ejs -v`

手動檢視資料夾 `node_modules`

# 透過yarn安裝樣板引擎

1. 請先到 <https://classic.yarnpkg.com/zh-Hant/>
2. 請點選「安裝YARN」，下載並且安裝
3. 在終端機視窗輸入  
**yarn add ejs**  
透過yarn安裝樣板引擎



12-4

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

1. 請先到 <https://classic.yarnpkg.com/zh-Hant/>
2. 請點選「安裝YARN」。
3. 網站會依據你的作業系統自動提示可以裝的版本。
4. 下載並且安裝
5. 在 Visual Studio Code 按下「Ctrl + 滑鼠右鍵」開啟 terminal 終端機視窗
6. 在專案根目錄底下，透過 yarn 安裝 ejs，指令: `yarn add ejs`

【Key Points】：

安裝YARN

在 Visual Studio Code 按下「Ctrl + 滑鼠右鍵」開啟 terminal 終端機視窗  
輸入 `yarn add ejs`，透過yarn安裝樣板引擎

# express-generator專案產生器

- express-generator是專案產生器，因此，安裝在全域。
- 安裝指令:  
**npm i express-generator -g**
- 安裝後請確認express-generator版本:  
**express --version**
- express-generator 可以自動幫你產生一個 express 專案，且透過參數就能安裝 view engine
- 產生專案的語法:  
**express -e {專案名稱}**  
-e代表指定 ejs view engine

12-5



- express-generator是專案產生器，因此，安裝在全域。
- 安裝指令:  
**npm i express-generator -g**
- 安裝後請確認express-generator版本:  
**express --version**
- express-generator 可以自動幫你產生一個 express 專案，且透過參數就能安裝 view engine
- 產生專案的語法:  
**express -e {專案名稱}**  
-e代表指定 ejs view engine

【Key Points】：

express-generator是專案產生器

安裝: **npm i express-generator -g**

產生專案的語法: **:express -e {專案名稱}**

# 以 git clone 的方式安裝 ejs

1. 在 package.json · dependencies 登記的 ejs 屬性值是 ejs 的 github repository 網址
2. 以 npm install 指令安裝 執行 npm install 指令時，npm 會 從指定的 github 網址下載 ejs 套件
  - 安裝套件的方式，除了透過套件管理工具，還可以指定 github 網址
  - 套件如果是開源套件，就能用這種指定網址的方式取得

```
① package.json > ...
1  [
2    "name": "tmp-02",
3    "version": "0.0.0",
4    "private": true,
5    "scripts": {
6      "start": "node ./bin/www"
7    },
8    "dependencies": {
9      "cookie-parser": "~1.4.4",
10     "debug": "~2.6.9",
11     "ejs": "https://github.com/tj/ejs.git",
12     "express": "~4.16.1",
13     "http-errors": "~1.6.3",
14     "morgan": "~1.9.1"
15   }
16 ]
17 }
```

12-6



1. 在 package.json · dependencies 登記的 ejs 屬性值是 ejs 的 github repository 網址
2. 執行 npm install 指令時，npm 會 從指定的 github 網址下載 ejs 套件
  - 安裝套件的方式，除了透過套件管理工具，還可以指定github網址。
  - 套件如果是開源套件就能用這種指定網址方式取得。
  - 指定git repository有個好處是，有時候專案需要裝的是套件的某個git tag 或version，就可以選擇用指定git url方式。
  - 使用npm 或yarn的方式還是比較好的，因為npm或yarn都有cache機制，可以加快安裝。

## 【Key Points】：

在 package.json · dependencies 登記的 ejs 屬性值是 ejs 的 github repository 網址  
以 npm install 指令安裝  
如果是開源套件就能用這種指定網址方式取得

## 12-2: 設定 EJS

- 什麼是樣板引擎
- EJS樣板引擎
- EJS 設定
- 如何傳遞參數到頁面樣板



designed by freepik



designed by freepik

12-7

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

- 樣板引擎：是一個工具，能將頁面樣板和要顯示的資料結合起來，生成 HTML 頁面。
- HTML頁面可以是靜態的。預先編寫好的 HTML 頁面就是靜態頁面。
- HTML頁面亦可以是動態，但動態HTML頁面需要配合樣板引擎產生。
- EJS是樣板引擎的一種，也是本課程使用的樣板引擎。它使用起來十分簡單，而且與 express 整合良好。
- ejs、hbs、handlebars、jade、pug 算是比較知名的view engine，其他還有twig、vash、dust、hjs等。

### 【Key Points】：

EJS樣板引擎

EJS 設定

如何傳遞參數到頁面樣板

# 什麼是樣板引擎

- 什麼是樣板引擎 ( Template Engine )
- 樣板板引擎和動態網頁的關係



12-8

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

- 樣板引擎：是一個工具，能將頁面樣板和要顯示的資料結合起來，生成 HTML 頁面。
- HTML頁面可以是靜態的。預先編寫好的HTML頁面就是靜態頁面。
- 而HTML頁面亦可以是動態，但動態HTML頁面需要配合樣板引擎產生。
- 樣板引擎的功能，是將頁面樣板和要顯示的資料結合起來生成 HTML 頁面。
- EJS 既可以運行在伺服器端，也可以執行在用戶端。大多數時候，它都用於伺服器端解析 HTML，解析完成後再傳輸給用戶端，就是最終瀏覽器所看見的資訊。

## 【Key Points】：

HTML頁面可以是靜態的。預先編寫好的HTML頁面就是靜態頁面。

樣板引擎的功能，是將頁面樣板和要顯示的資料結合起來生成 HTML 頁面。

EJS 既可以運行在伺服器端，也可以執行在用戶端。大多數時候，它都用於伺服器端解析 HTML。

# EJS樣板引擎

- EJS是樣板引擎的一種
- 內嵌式的樣板引擎，可以將邏輯與內容直接嵌入到HTML頁面上
- 只要會基本的HTML就可以容易學會
- 使用起來十分簡單，而且與 express 整合良好
- 官方網站網址：

<https://ejs.co/>



12-9

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

- EJS是樣板引擎的一種，也是本課程使用的樣板引擎。它使用起來十分簡單，而且與 express 整合良好。
- EJS的全名是「Embedded Javascript」，顧名思義就是內嵌式的樣板引擎，可以將邏輯與內容直接嵌入到HTML頁面上。
- EJS好處是只要會基本的HTML就可以容易學會
- 而EJS可以同時運行在用戶端和伺服器端，用戶端安裝直接引入文件即可，伺服器端用npm包安裝
- EJS簡單容易使用，是很受歡迎的樣板引擎。

## 【Key Points】：

EJS是樣板引擎的一種

內嵌式的樣板引擎，可以將邏輯與內容直接嵌入到HTML頁面上

只要會基本的HTML就可以容易學會

# EJS 設定

- 修改 app.js
- 設定頁面引擎
- 設定頁面渲染

```
app.set('view engine', 'ejs');

app.get('/', function(req, res) {
  res.render('default', {
    title: '首頁',
    users: ['Kai', 'aYen', 'Kyousuke']
  });
});
```

- 創建頁面樣板 ./views/default.ejs

```
<% for(var i=0; i<users.length; i++) {%
  <li><%= users[i] %></li>
<% } %>
```

12-10



1. 首先，宣告 view engine 為 EJS
2. 然後，設定樣板名稱，以本例來說，default 就是我們的樣板檔案名稱
3. 接下來，將「網頁標題(title)與其他內容(users)」等資料打包成物件，透過 res.render() 傳給 EJS，稍後由 EJS 將資料嵌入到網頁之中。
4. 創建一個新樣板，檔名叫 default.ejs 。
5. 在 default.ejs 使用 EJS 標籤<%= %>轉譯資料成HTML內容 。

## 【Key Points】：

設定頁面引擎

傳送 View Data

<%= %>

# 如何傳遞參數到頁面樣板

- **default.ejs**

```
<html>
  <body>
    <h1><%= title %></h1>
    <p><%= description %></p>
  </body>
</html>
```

## app.js

```
app.get('/', function(req, res){
  //使用default樣板，並可以帶入參數給樣板利用
  res.render('default',{
    title: 'EJS 樣板測試',
    description: '我是EJS樣板'
  });
});
```



12-11



以 `res.render()` 渲染 (動態輸出) 出 HTML 內容：

```
res.render('default',{
  title: 'EJS 樣板測試',
  description: '我是EJS樣板'
});
```

寫在大括號內是樣板參數，也叫 View Data (給 View 的 Data)，以本例來說，內含標題及描述 (title、description)，帶入資給樣板：`default.ejs`。

EJS的語法以 `<%>` 和 `>%<` 包夾而成標籤，標籤裡頭的內容都是動態產生。`default.ejs`的內容：

```
<html>
  <body>
    <h1><%= title %></h1>
    <p><%= description %></p>
  </body>
</html>
```

使用EJS標籤`<h1><%= title %></h1>` 渲染 出 HTML 內容: `<h1>EJS 樣板測試'</h1>`

### 【Key Points】：

以 `res.render()` 渲染 (動態輸出) 出 HTML 內容

寫在大括號內是樣板參數，也叫 View Data (給 View 的 Data)

EJS的語法以 `<%>` 和 `>%<` 包夾而成標籤，標籤裡頭的內容都是動態產生

# 12-3:EJS Tags

- 關於EJS 標籤
- 常用EJS標籤
- 基本頁面元素
- 流程控制標籤



designed by freepik



designed by freepik

12-12

III 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

- EJS標籤(Tags)是動態處理網頁內容的重要工具
  - 在 <% %> 內的都視同是程式
  - 由<%= %>標籤裡頭的程式動態產生內容
- <%= %> 與 <%- %> 有何差別

- 頁面可以使用 JavaScript 進行流程控制:

```
<% if ( ) { %> ... <% } %>
<% for ( ) { %> ... <% } %>
```

## 【Key Points】：

在 <% %> 內的都視同是程式

<%= %> 與 <%- %> 有何差別

頁面可以使用 JavaScript 進行流程控制

# 關於EJS 標籤

- 什麼是EJS標籤

- <%      %>
- <%=     %>
- <%-    %>

```
<html>
  <body>
    <h1><%= title %></h1>
    <p><%= description %></p>
  </body>
</html>
```

EJS Template

Website:<http://www.embeddedjs.com/>

12-13



- EJS標籤([Tags](#))是動態處理網頁內容的重要工具
- 在 <% %> 內的都視同是程式
- 由<%= %>標籤裡頭的程式動態產生內容
- 通過不同標籤，EJS 將生成不同類型的HTML元素

<%= %> 與 <%- %> 的差別舉例說明如下：

假設 data 是一個字串，內容是 "Line1<br>Line2"

以 <%= data %> 會動態生成出：「Line1&lt;br&gt;Line2」，瀏覽器收到內容後，會顯示成：Line1<br>Line2

以 <%- data %> 會動態生成出：「Line1<br>Line2」，瀏覽器收到內容後，會顯示成兩行。

簡單地說，<%- %>會渲染出 HTML 標籤，而 <%= %> 則會轉換 < 符號為 &lt;

## 【Key Points】：

<% %> 內的都視同是程式  
<%- %>會渲染出 HTML 標籤  
而 <%= %> 會轉換小於、大於符號

## 常用EJS標籤

- <%      %>            流程控制標籤
- <%=     %>          輸出標籤（必要時，轉譯H T M L 符號）
- <%-     %>          輸出標籤（不會轉譯H T M L 符號）
- <%#     %>          注釋標籤

12-14



- <% %>作為內崁 JavaScript 的關鍵符號，也就是說放在這中間的部分就會被視為 JavaScript 來執行
- <%= %>裡面的 JavaScript 變數，則會以 `toString()` 的方式將其轉換為字串，並加入至網頁中。
- 例子:輸出變數

```
title = "EJS"
<p><%= title %></p>
<p>EJS</p>
```
- <%- %>HTML 輸出「非轉譯的內容」(<br>不會轉譯為 &lt;br&gt;)，通常用於合併樣板並輸出至頁面。
- 例子:合併header.ejs及footer.ejs

```
<%- include('header') %>
HTML code
<%- include('footer') %>
```
- <%# %>注釋標籤。不用執行、不必輸出內容。

### 【Key Points】：

- <% %> 流程控制標籤
- <%= %> 輸出標籤（必要時，轉譯H T M L 符號）
- <%- %> 輸出標籤（不會轉譯H T M L 符號）

# 基本頁面元素

- HTML 標籤(head, body)
- 網頁的標頭區 (**header**)
- 網頁的導覽列 (**nav**)
- 網頁的主要內容 (**body**)
- 網頁的頁尾 (**footer**)

```
<html>
  <header></header>
  <body>
    <nav></nav>
    <content></content>
  </body>
</html>
```

12-15



- 在 html 中，以 "<" 開頭並且以 ">" 結尾的稱之為標籤(Tag) ，並且有起始標籤與結束標籤，如 "<div>" 是起始標籤， "</div>" 是結束標籤。
- 在網頁結構中，通常包括以下元素：網頁的標頭，網頁的導覽列，網頁的主要內容，網頁的頁尾
- 標頭會以 head 標籤作開頭及結尾，當中包含HTML標題定義、外部、指令碼代碼、格式代碼等處理資訊。
- 導覽列主要作用是作為橋樑連結各頁面，通常會以 nav 標籤作開頭及結尾。導覽列會包含個頁面的連結。
- 而頁面主要內容會放在body標籤中
- 網頁的頁尾則以footer標籤作開頭及結尾，通常放置聯絡方式、著作權宣告等等。

## <Note>

針對固定不變的內容，例如頁尾，可考慮切分出去成一個獨立檔案，然後，再以 <%- include( "footer" ) %> 嵌進來

## 【Key Points】：

網頁的導覽列 (nav)  
網頁的主要內容 (body)  
網頁的頁尾 (footer)

# 流程控制標籤

```
app.get('/', function(req, res){  
    //使用default樣板，並可以帶入參數給樣板利用  
    res.render('default',{  
        title: 'EJS 樣板測試',  
        items: ['我是', '我是二', '我是三', '我是四', '我是五']  
    });  
});
```

```
<!-- //title變數有資料則印出，無的話使用預設文字 -->  
<% if (title) { %>  
    <h1><%= title %></h1>  
<% } else { %>  
    <h1>預設文字</h1>  
<% } %>  
  
<!-- //印出items陣列中所有數值 -->  
<ul>  
    <% for(var i=0; i<items.length; i++) { %>  
        <li><%= items[i] %></li>  
    <% } %>  
</ul>
```



## EJS 樣板測試

- 我是一
- 我是二
- 我是三
- 我是四
- 我是五

12-16



EJS標籤的其中一項功能，是讓頁面可以使用 JavaScript，頁面得以根據流程控制，動態輸出內容。

本課程的其他內容，會有實例示範如何使用 if (登入控制) 及 for 流程控制(顯示資料庫內容)。先舉一個簡單的例子示範語法：

```
<% if (title) { %>  
<h1><%= title %></h1>  
<% } else { %>  
<h1>預設文字</h1>  
<% } %>
```

以上述程式來說，如果 title 有資料就輸出資料；否則就輸出預設文字。透過流程控制標籤，可以偵測變數是否存在，達致控制輸出的效果。

另外，亦可透過控制標籤及使用 for 迴圈來輸出陣列中的多項資料：

```
<% for(var i=0; i<items.length; i++) { %>  
    <li><%= items[i] %></li>  
<% } %>
```

### 【Key Points】：

頁面可以使用 JavaScript 進行流程控制

```
<% if ( ) { %> ... <% } %>  
<% for ( ) { %> ... <% } %>
```

# Summary〈精華回顧〉

- EJS是最受歡迎的樣板引擎，簡單易用、與express整合良好
- 配合樣板引擎可以產生動態頁面
- 安裝 EJS 指令是 `npm install ejs`
- 設定頁面引擎：`app.set('view engine', 'ejs');`
- 設定頁面渲染：`app.get('/', function(req, res){  
 res.render('default', {  
 title: 'EJS 樣板測試'  
 });  
});`
- 輸出內容：`<h1><%= title %></h1>`
- 流程控制標籤：`<% if ( ) { %>...<% } %>    <% for ( ) { %> ...<% } %>`



12-17

 財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

EJS是最受歡迎的樣板引擎，簡單易用、與express整合良好  
配合樣板引擎可以產生動態頁面  
安裝 EJS 指令是 `npm install ejs`

設定頁面引擎：`app.set('view engine', 'ejs');`

設定頁面渲染：`app.get('/', function(req, res){  
 res.render('default', {  
 title: 'EJS 樣板測試'  
 });});`

輸出內容：

`<h1><%= title %></h1>`

流程控制標籤：

`<% if ( ) { %>...<% } %>    <% for ( ) { %> ...<% } %>`

## 【Key Points】：

安裝與設定 EJS引擎

輸出內容：`<h1><%= title %></h1>`

流程控制標籤：`<% if ( ) { %>...<% } %>    <% for ( ) { %> ...<% } %>`

# 線上程式題

- **12-1 傳遞參數給 EJS**

請修改程式碼，使其能夠正常render。

- **12-2 以 EJS 輸出 HTML Tag**

請修改程式碼，以 EJS 輸出 HTML Tag 紿瀏覽器。

- **12-3 在 EJS 使用 for 迴圈**

請修改一下程式碼使畫面能夠正常運行

Estimated time:

20 min.

12-18



題目名稱: 12-1 傳遞參數給 EJS

內容說明:

請修改程式碼，使其能夠正常render。

題目名稱: 12-2 以 EJS 輸出 HTML Tag

內容說明:

請修改程式碼，以 EJS 輸出 HTML Tag 紿瀏覽器。

題目名稱: 12-3 在 EJS 使用 for 迴圈

內容說明:

請修改一下程式碼使畫面能夠正常運行

請閱讀教學系統的程式與說明，然後，到「// 作答區」填入答案。

答案有區分大寫小寫。

【Key Points】：

12-1 傳遞參數給 EJS

12-2 以 EJS 輸出 HTML Tag

12-3 在 EJS 使用 for 迴圈

# 課後練習題(Lab)

- **情節描述:**

本Lab 是假設在已經安裝Node.js環境的Windows 電腦下，練習如何安裝EJS模塊、繫結資料，最後創建多重EJS頁面模塊並且整合在一起。

- **預設目標:**

- 學會安裝EJS模塊
- 繫結資料
- 創建多重EJS頁面模塊並且整合在一起。

- **Lab01: 如何安裝及設定**

Estimated time:

20 minutes

- **Lab02: 創建多重EJS頁面模塊並且整合在一起**

12-19



## 【情節描述】

本Lab 是假設在已經安裝Node.js環境的Windows 電腦下，練習如何安裝EJS模塊、繫結資料，最後創建多重EJS頁面模塊並且整合在一起。

## 【預設目標】

學會安裝EJS模塊、繫結資料、創建多重EJS頁面模塊並且整合在一起。

關鍵程式:

```
res.render('default',{
    title: 'EJS 樣板測試'
});
```

```
<h1><%= title %></h1>
```

```
<%- include('header'); -%>
```

完成後的程式與檔案，請參考 Example 資料夾的內容。

## 【Key Points】:

```
res.render()
<h1><%= title %></h1>
<%- include('header'); -%>
```

# 範例程式使用說明

- 範例程式資料夾: Module\_12\_example
- 使用步驟:
  1. 安裝 Node.js
  2. 安裝 Visual Studio Code
  3. 以 Visual Studio Code 開啟本模組的範例資料夾
  4. 在 Visual Studio Code 按下「Ctrl + 滑鼠右鍵」開啟 terminal 終端機視窗。
  5. 在終端機視窗，輸入下列指令，安裝必要的模組套件:  
npm install express ejs
  6. 在終端機視窗，輸入 node index.js
  7. 啟動瀏覽器，連接 http://localhost:5000

12-20



使用步驟:

1. 安裝 Node.js  
<https://nodejs.org/en/>
2. 安裝 Visual Studio Code  
<https://code.visualstudio.com/>
3. 以 Visual Studio Code 開啟範例資料夾  
在檔案總管，滑鼠右鍵點按「本範例資料夾」，從快捷功能表點選「以Code開啟」  
或者，啟動 Visual Studio Code 之後，功能表 File | Open Folder，選擇「本範例資料夾」
4. 在 Visual Studio Code 按下「Ctrl + 滑鼠右鍵」開啟 terminal 終端機視窗。
5. 在終端機視窗，輸入下列指令，安裝必要的模組套件:  
npm install express ejs
6. 在終端機視窗，輸入 node index.js
7. 啟動瀏覽器，連接 http://localhost:5000

【Key Points】：

程式執行環境 Node.js

程式開發編輯環境: Visual Studio Code

在 Visual Studio Code 按下「Ctrl + 滑鼠右鍵」開啟 terminal 終端機視窗，輸入：「node 主程式.js」