

程式人



用十分鐘瞭解 JavaScript 的模組

《還有關於 npm 套件管理的那些事情》

陳鍾誠

2016 年 3 月 29 日

話說

- 如果你聽到 node.js, npm, gulp, grunt, bower, yo, browserify, webpack 等名詞，應該會以為這是某外星人說的《克林貢語》吧！

雖然距離我第一次遇見 JavaScript

- 已經有 20 個年頭了！

- 但我還是不太瞭解

JavaScript 這個語言

而且自從 node. js 出現之後

- 我就不瞭解它了！

即使

- 我開始用 node.js 已經超過 3 年了
- 我仍然一直聽到各種《克林貢語》出現在我的耳邊！
- 彷彿我來到這個星球才只有三天而已

更別說是

- 前端領域變來變去的套件，像是 angular, react, backbone, ember 等等那些火星文了！

沒辦法

- 自從我們揮別了《微軟帝國》的統一世界之後，世界就開始崩解、崩解、再崩解...

於是我們有了

- Apple, iPod, iPad, iPhone, iMac, iOS, iTune, Xcode, ...
- Google, Android, GAE, Play ...
- Amazon, EC2, Arduino, Raspberry Pi, Ruby, Python, Haskell ...

即使只把焦點鎖定在 node.js 領域

- 我們也有 async, lodash, underscore, async, kama, mocha, cordova, forever, express, koa, connect, commander, ...
等等數十萬種套件可以使用

等等、還沒完

- 由於這張投影片太小，無法盡列，所以請自行上 npm 去找到這些套件！

npm ?


Browser address bar: <https://www.npmjs.com>


Navigation links: Navy Penguin Mariachi, npm On-Site, npm Private Packages, npm Open Source, documentation, support


Search bar: find packages


Sign up or log in

npm is the package manager for nodebots







 260,169
total packages

 45,462,317
downloads in the last day

 859,197,758
downloads in the last week

 3,686,536,515
downloads in the last month

packages people 'npm install' a lot

 browserify browser-side require() the node way 13.0.0 published 3 months ago by feross	 express Fast, unopinionated, minimalist web... 4.13.4 published 2 months ago by doug..	 pm2 Production process manager for Nod.. 1.0.0 published 3 months ago by tknew
 grunt-cli The grunt command line interface. 0.1.13 published 2 years ago by tkellen	 npm a package manager for JavaScript 3.7.1 published 2 months ago by iarna	 karma Spectacular Test Runner for JavaScri.. 0.13.19 published 3 months ago by dign..

那是甚麼？

喔！

- 那是 node.js 裏的一個指令
- 是 node package manager 的簡稱

npm 可以用來幹嘛？

喔！

- 可以用來

- 安裝套件

- 發佈套件

- 管理套件

問題是

- 甚麼是 npm 的套件？
- 我要怎麼使用 npm 的套件？
- 我要怎麼撰寫 npm 的套件？

喔！

- 這些問題比較複雜！
- 讓我們先從《使用套件》開始講起！

想使用 npm 套件

- 首先您得上 npm 網站去查詢
- 你想要用的套件！

舉例而言

- 假如我們知道有個叫 numeric 的套件
- 而且知道這是一個矩陣運算的套件
- 那麼我們就可以上 npm 網站去看看
這個套件的描述！

您可以搜尋該套件

Browser address bar: <https://www.npmjs.com>

Navigation links: Navy Penguin Mariachi, npm On-Site, npm Private Packages, npm Open Source, documentation, support

Search bar: **numeric**

Search results:

- numeric**
Numerical analysis in javascript
- numerical**
gives yr numbers math helpers
- numerics**
Client library for the numerics.io custom metrics API
- numeric-buffer**
Create Node.js Buffers with integers
- numeric_arrays**
Provides a library of utility mixins for dealing with numeric arrays in JavaScript (without extending prototypes)
- numeric array**
- numerical**
- numeric data**
- numeric arrays**

Footer:

- grunt-cli**
The grunt command line interface.
0.1.13 published 2 years ago by tkellen
- npm**
a package manager for JavaScript
3.7.1 published 2 months ago by iarna
- karma**
Spectacular Test Runner for JavaScript
0.13.19 published 3 months ago by dign...

找到之後

The screenshot shows a web browser window with the URL `https://www.npmjs.com/search?q=numeric`. The browser's address bar shows the site is `npm Inc. [US]`. The page has a red header with navigation links: `Neverending Perpetual Motion`, `npm On-Site`, `npm Private Packages`, `npm Open Source`, `documentation`, and `support`. Below the header is a dark grey bar containing the `npm` logo, a search bar with the text `numeric`, a magnifying glass icon, and a link to `sign up or log in` next to a small cat icon. The main content area displays the search results for `numeric`, showing 1975 results. The first three results are listed below:

- numeric** clay
Numerical analysis in javascript
★ 1 v1.2.6
numeric, analysis, math
- numeric-pattern** bendrucker
The numeric pattern (triggers the numpad on mobile)
★ 0 v1.0.0
input, numeric, pattern, numpad, number, mobile, html
- obj-numeric** bendrucker
Create an object with incremented numeric values from an array of keys
★ 0 v1.0.0
object, obj, numeric, array, keys

On the right side of the results, there is a link to `lob.com`, `PaperG`, `Edyn` and lots of other companies are hiring javascript developers. View all 34...

看看套件的描述

The screenshot shows the npm website interface for the 'numeric' package. The browser's address bar displays the URL 'https://www.npmjs.com/package/numeric'. The page features a red navigation bar with links to 'Nettle Pie Market', 'npm On-Site', 'npm Private Packages', 'npm Open Source', 'documentation', and 'support'. Below this is a dark search bar with the 'npm' logo, a search input field containing 'find packages', a search icon, and a 'sign up or log in' button with a cat icon. The main content area for the 'numeric' package includes a 'public' badge, a star icon, and the title 'Numerical analysis in javascript'. The description states that Numeric Javascript is a browser-based library for numerical analysis. It also provides links to the official website and a Google Group for discussion. The 'License' section mentions the MIT license. On the right sidebar, there is a promotion for 'npm On-Site', a command to install the package, a link to learn more, and download statistics showing 76 downloads in the last day and 1,700 in the last week.

← → ↻ **npm Inc. [US]** <https://www.npmjs.com/package/numeric> ☆

Nettle Pie Market npm On-Site npm Private Packages npm Open Source documentation support

npm find packages 🔍 sign up or log in 🐱

numeric public ★

Numerical analysis in javascript

Numeric Javascript is a javascript library for doing numerical analysis in the browser. Because Numeric Javascript uses only the javascript programming language, it works in many browsers and does not require powerful servers.

Numeric Javascript is for building "web 2.0" apps that can perform complex calculations in the browser and thus avoid the latency of asking a server to compute something. Indeed, you do not need a powerful server (or any server at all) since your web app will perform all its calculations in the client.

For further information, see <http://www.numericjs.com/> Discussion forum: <http://groups.google.com/group/numericjs>

License

Numeric Javascript is copyright by Sébastien Loisel and is distributed under the MIT license. Details

npm On-Site

Private npm for enterprises. [Start a free trial »](#)

📦 npm install numeric

[how? learn more](#)

1.2.6 is the latest of 9 releases

github.com/sloisel/numeric

Stats

76 downloads in the last day

1,700 downloads in the last week

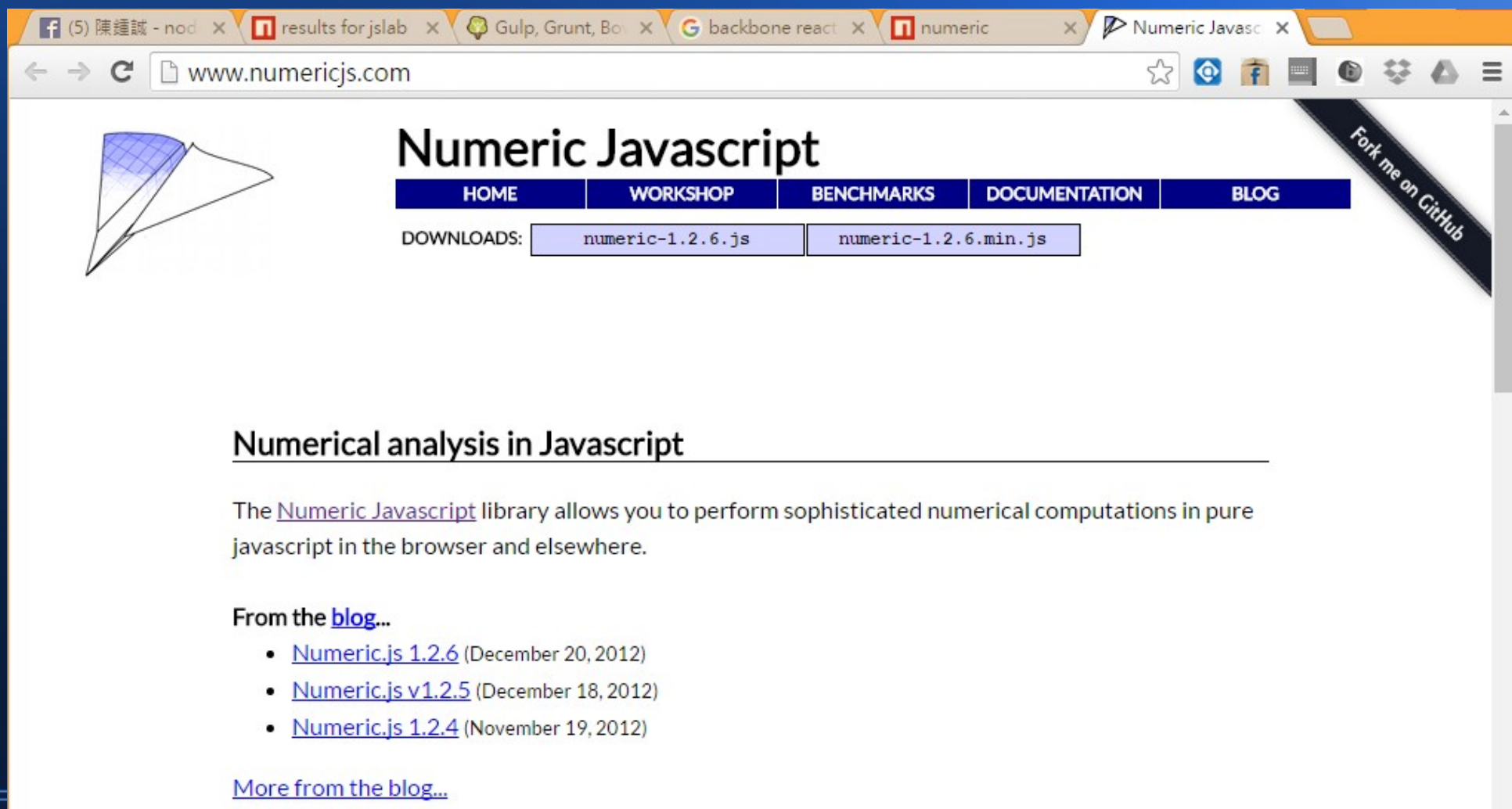
有些比較好的套件

- 還會有官方網站！

For further information, see <http://www.numericjs.com/> Discussion forum:
<http://groups.google.com/group/numericjs>

於是你可以去看看

- 該套件的詳細內容！



The screenshot shows a web browser window with the URL www.numericjs.com. The page features a logo on the left, a navigation bar with links to HOME, WORKSHOP, BENCHMARKS, DOCUMENTATION, and BLOG, and a section for downloads with buttons for `numeric-1.2.6.js` and `numeric-1.2.6.min.js`. A diagonal banner on the right says "Fork me on GitHub". Below the navigation bar, the page title "Numerical analysis in Javascript" is underlined. The main text describes the library as a tool for performing numerical computations in pure javascript. A section titled "From the blog..." lists three recent blog posts with their dates. At the bottom, there is a link to "More from the blog...".

Numeric Javascript

HOME WORKSHOP BENCHMARKS DOCUMENTATION BLOG

DOWNLOADS: `numeric-1.2.6.js` `numeric-1.2.6.min.js`

Numerical analysis in Javascript







The [Numeric Javascript](#) library allows you to perform sophisticated numerical computations in pure javascript in the browser and elsewhere.

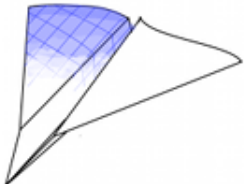
From the [blog...](#)

- [Numeric.js 1.2.6](#) (December 20, 2012)
- [Numeric.js v1.2.5](#) (December 18, 2012)
- [Numeric.js 1.2.4](#) (November 19, 2012)

[More from the blog...](#)

以及使用說明

← → ↻ www.numericjs.com/workshop.php ☆      



Numeric Javascript

HOME	WORKSHOP	BENCHMARKS	DOCUMENTATION	BLOG
------	----------	------------	---------------	------

DOWNLOADS: [numeric-1.2.6.js](#) [numeric-1.2.6.min.js](#)

[START OVER](#) [MAKE PERMALINK](#) Version: [lib/numeric-1.0.0.js](#) (Update to [lib/numeric-1.2.6.js](#))

IN> `// Linear algebra example. We start with a matrix.
A = [[1,2,3],
 [4,5,6],
 [7,3,9]]`

OUT> `[[1, 2, 3],
 [4, 5, 6],
 [7, 3, 9]]`

IN> `// Let's also make a vector.
x = [3,1,2];`

OUT> `[3, 1, 2]`

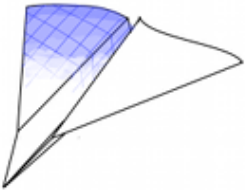
IN> `// Matrix-vector product.
b = numeric.dot(A,x);`

OUT> `[11, 29, 42]`

Fork me on GitHub

還有函式庫定義等等

← → ↻ www.numericjs.com/documentation.html ☆



Numeric Javascript

HOME WORKSHOP BENCHMARKS DOCUMENTATION BLOG

DOWNLOADS: [numeric-1.2.6.js](#) [numeric-1.2.6.min.js](#)

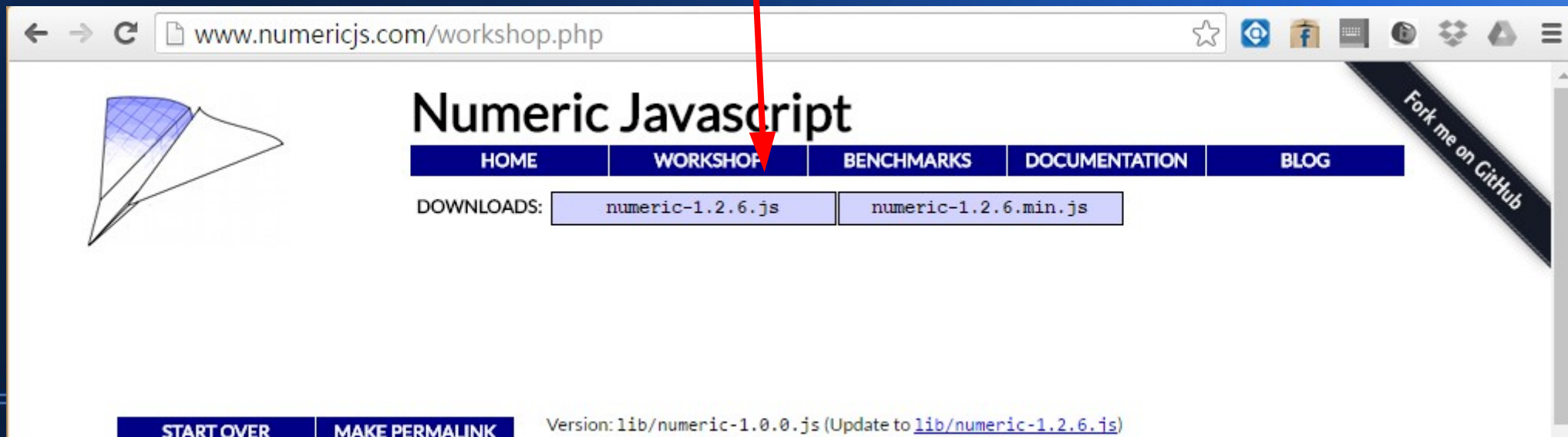
Fork me on GitHub

Reference card for the `numeric` module

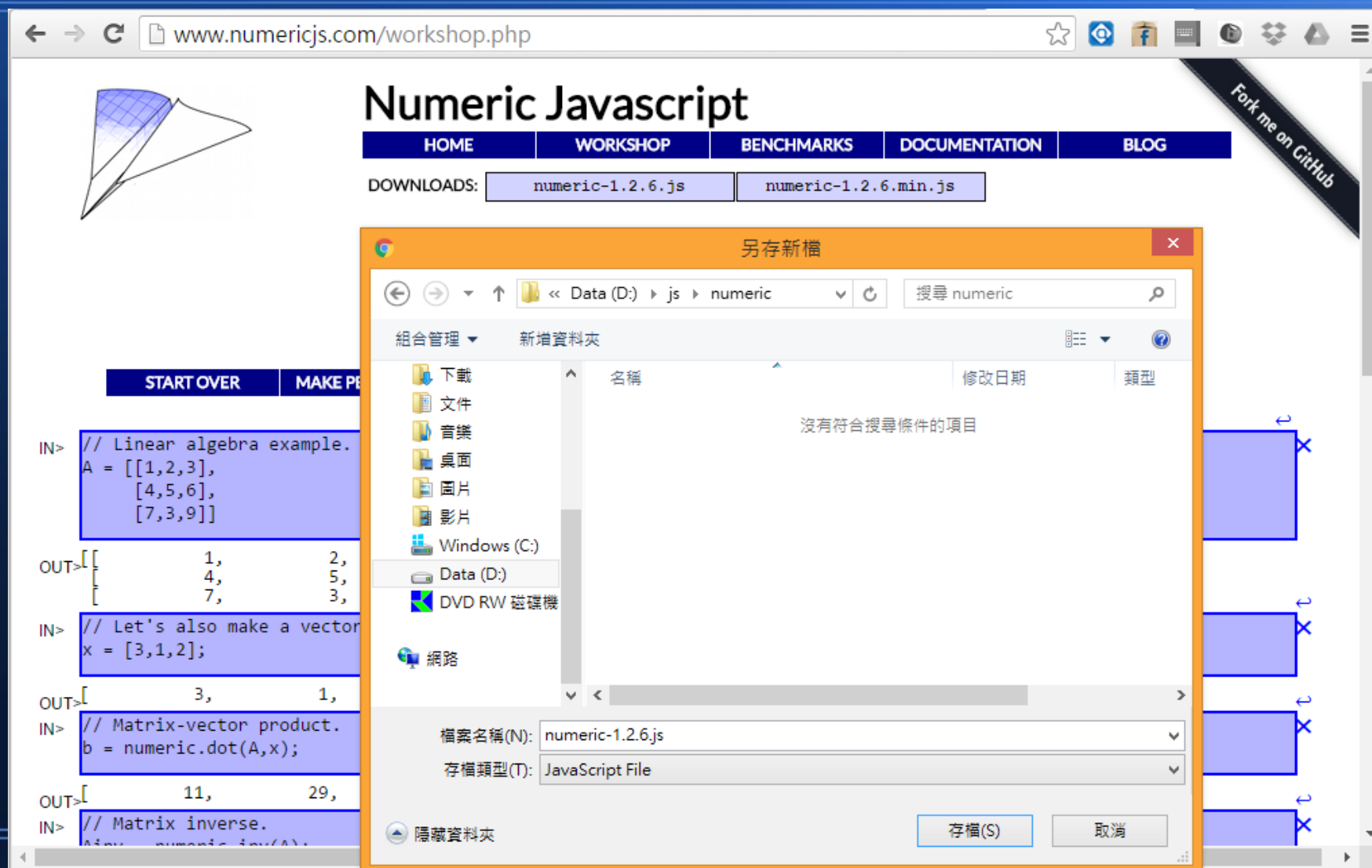
Function	Description	Function	Description	Function	Description
<code>abs</code>	Absolute value	<code>dot</code>	Matrix-Matrix, Matrix-Vector and Vector-Matrix product	<code>round</code>	Pointwise <code>Math.round(x)</code>
<code>acos</code>	Arc-cosine	<code>eig</code>	Eigenvalues and eigenvectors	<code>rrshift</code>	Pointwise <code>x>>>y</code>
<code>add</code>	Pointwise sum <code>x+y</code>	<code>epsilon</code>	2.220446049250313e-16	<code>rrshifteq</code>	Pointwise <code>x>>>=y</code>
<code>addeq</code>	Pointwise sum <code>x+=y</code>	<code>eq</code>	Pointwise comparison <code>x===y</code>	<code>rshift</code>	Pointwise <code>x>>y</code>
<code>all</code>	All the components of <code>x</code> are true	<code>exp</code>	Pointwise <code>Math.exp(x)</code>	<code>rshifteq</code>	Pointwise <code>x>>=y</code>
<code>and</code>	Pointwise <code>x && y</code>	<code>floor</code>	Pointwise <code>Math.floor(x)</code>	<code>same</code>	<code>x</code> and <code>y</code> are entrywise identical
<code>andeq</code>	Pointwise <code>x &= y</code>	<code>geq</code>	Pointwise <code>x>=y</code>	<code>seedrandom</code>	The <code>seedrandom</code> module
<code>any</code>	One or more of the components of <code>x</code> are true	<code>getBlock</code>	Extract a block from a matrix	<code>setBlock</code>	Set a block of a matrix
<code>asin</code>	Arc-sine	<code>getDiag</code>	Get the diagonal of a matrix	<code>sin</code>	Pointwise <code>Math.sin(x)</code>
		<code>gt</code>	Pointwise <code>x>y</code>	<code>solve</code>	Solve <code>Ax=b</code>

對於那些

- 可以直接掛進瀏覽器的前端套件而言
- 上面通常會有 `.js` 檔案的下載點！



您可以按右鍵選擇下載後存檔使用



The screenshot shows the 'Numeric Javascript' website's workshop interface. The browser address bar displays 'www.numericjs.com/workshop.php'. The website has a navigation menu with 'HOME', 'WORKSHOP', 'BENCHMARKS', 'DOCUMENTATION', and 'BLOG'. Below the menu, there are download links for 'numeric-1.2.6.js' and 'numeric-1.2.6.min.js'. A 'Fork me on GitHub' banner is visible in the top right corner.

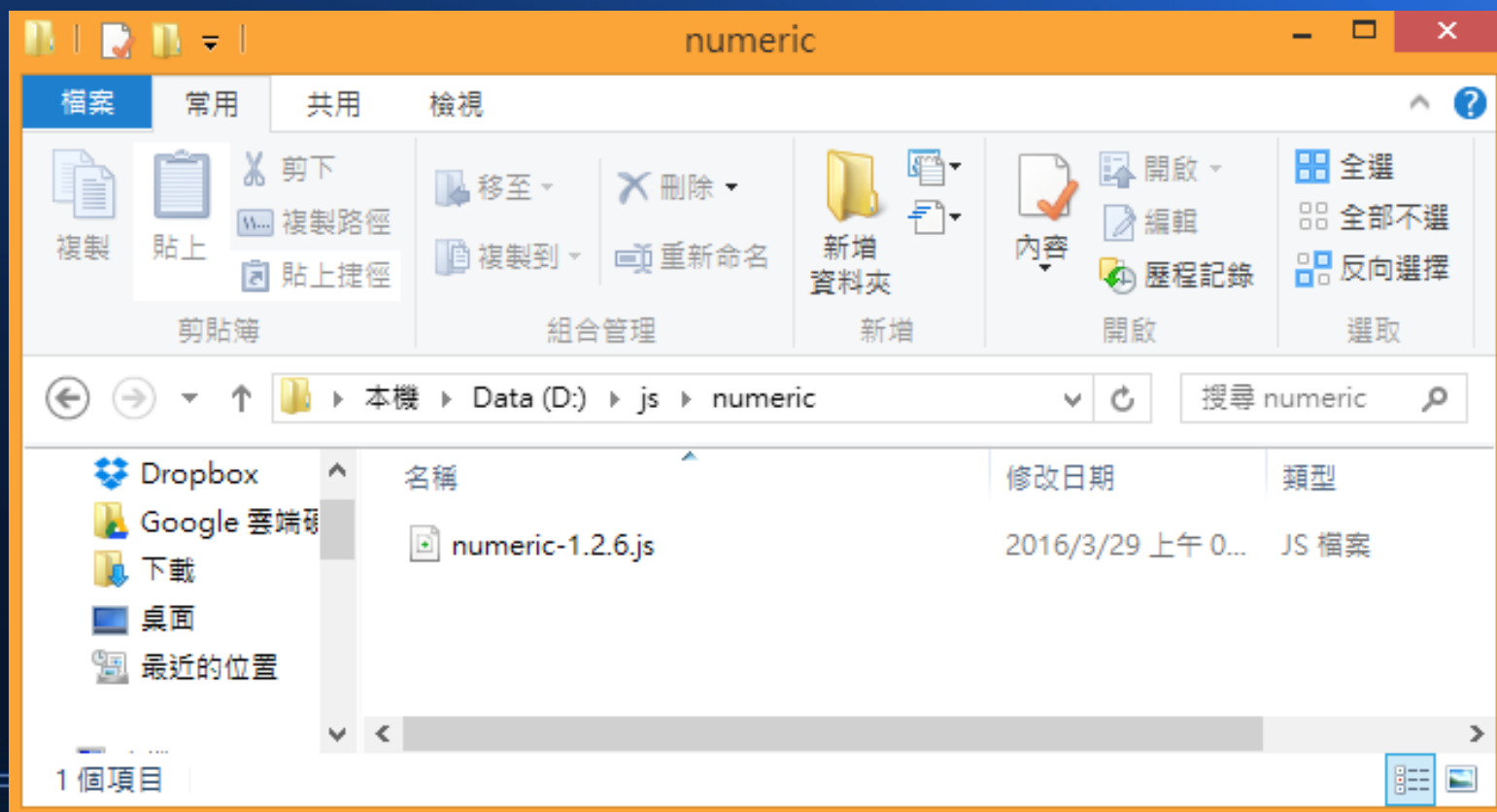
On the left side of the workshop, there is a code editor with the following content:

```
IN> // Linear algebra example.  
A = [[1,2,3],  
      [4,5,6],  
      [7,3,9]]  
  
OUT> [[ 1, 2,  
       4, 5,  
       7, 3, 9]]  
  
IN> // Let's also make a vector  
x = [3,1,2];  
  
OUT> [[ 3, 1,  
       4, 5,  
       7, 3, 9]]  
  
IN> // Matrix-vector product.  
b = numeric.dot(A,x);  
  
OUT> [[ 11, 29,  
       41, 52,  
       73, 84, 95]]  
  
IN> // Matrix inverse.  
inv = numeric.inv(A);
```

Overlaid on the bottom right of the workshop is a Windows file save dialog titled '另存新檔' (Save As). The dialog shows the current directory as 'Data (D:) > js > numeric'. The search bar contains 'numeric'. The file list is empty, displaying the message '沒有符合搜尋條件的項目' (No items match the search criteria). The '檔案名稱(N):' (File name) field contains 'numeric-1.2.6.js', and the '存檔類型(T):' (Save as type) is set to 'JavaScript File'. The '儲存(S)' (Save) button is highlighted.

但問題是

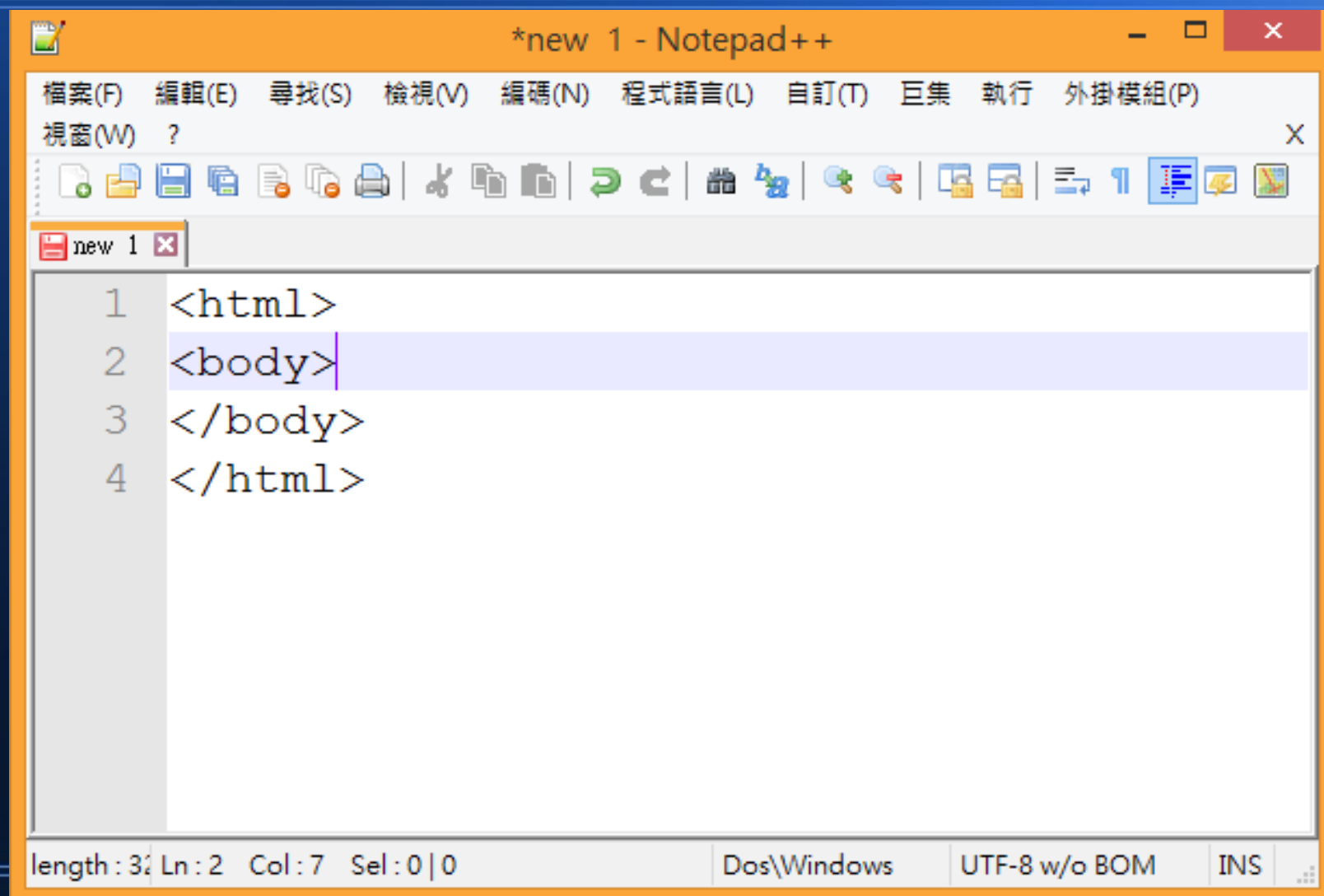
- 下載完 .js 檔案後要怎麼用呢？



對於這種 .js 檔案

- 通常是放在網頁內使用的！

您必須要先寫個 html 網頁

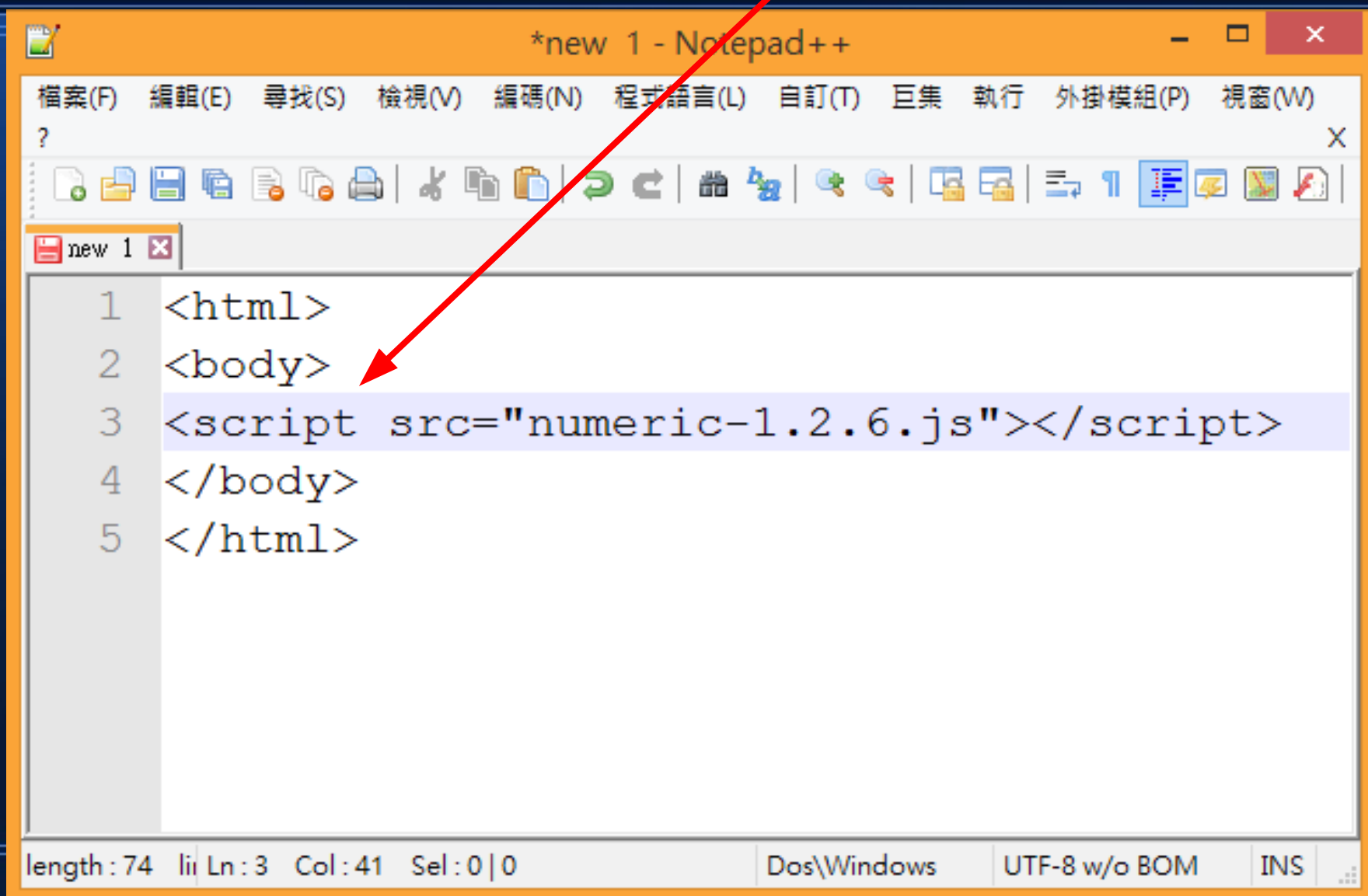


A screenshot of the Notepad++ application window. The title bar reads '*new 1 - Notepad++'. The menu bar includes '檔案(F)', '編輯(E)', '尋找(S)', '檢視(V)', '編碼(N)', '程式語言(L)', '自訂(T)', '巨集', '執行', and '外掛模組(P)'. The toolbar contains various icons for file operations and editing. The main text area shows a new document with the following HTML structure:

```
1 <html>
2 <body>
3 </body>
4 </html>
```

The status bar at the bottom displays 'length: 32 Ln: 2 Col: 7 Sel: 0 | 0', 'Dos\Windows', 'UTF-8 w/o BOM', and 'INS'.

記得加上 script 引用

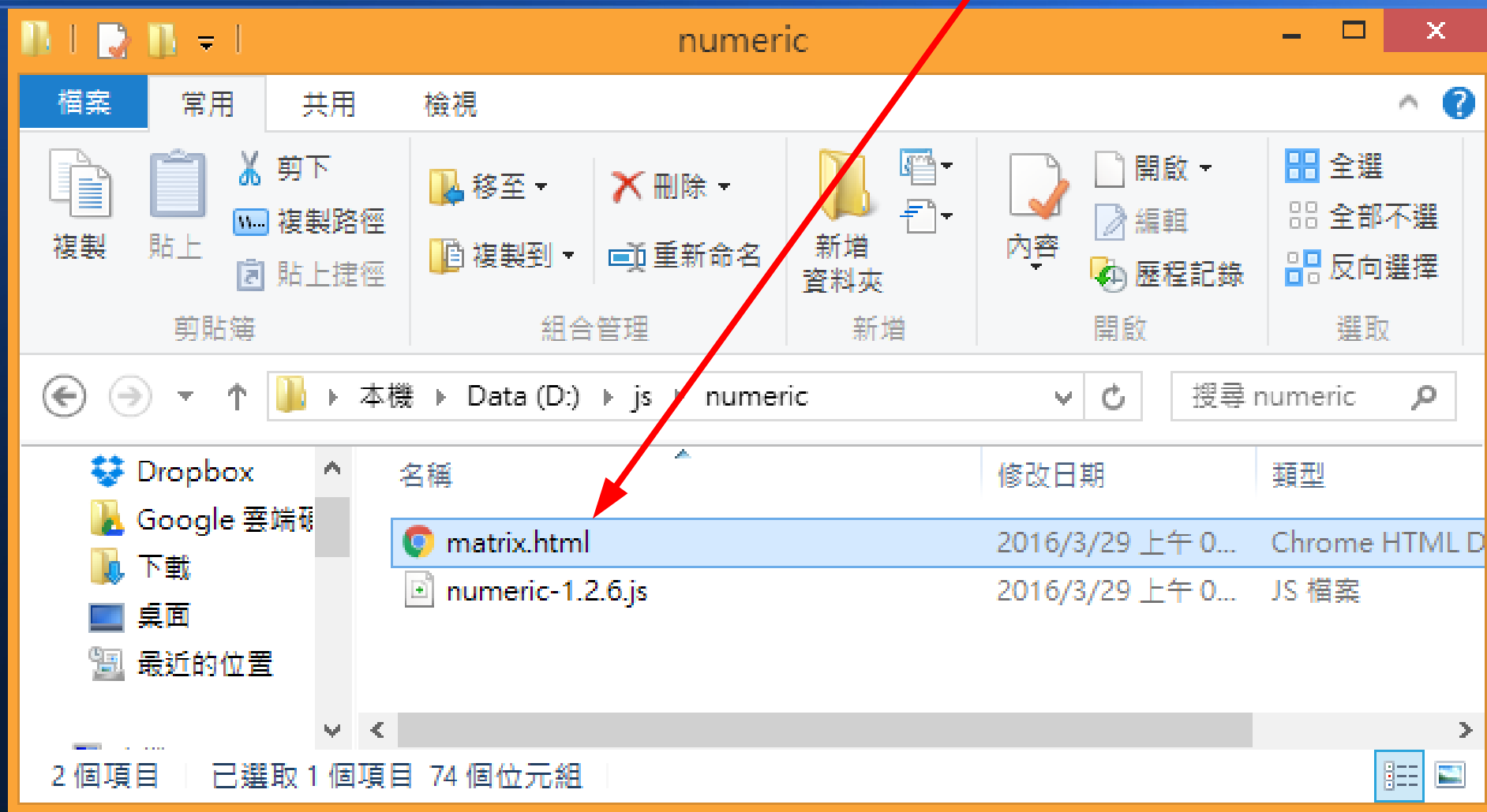


The screenshot shows the Notepad++ application window titled '*new 1 - Notepad++'. The menu bar includes 檔案(F), 編輯(E), 尋找(S), 檢視(V), 編碼(N), 程式語言(L), 自訂(T), 巨集, 執行, 外掛模組(P), and 視窗(W). The toolbar contains various icons for file operations and editing. The editor window shows a document with the following HTML code:

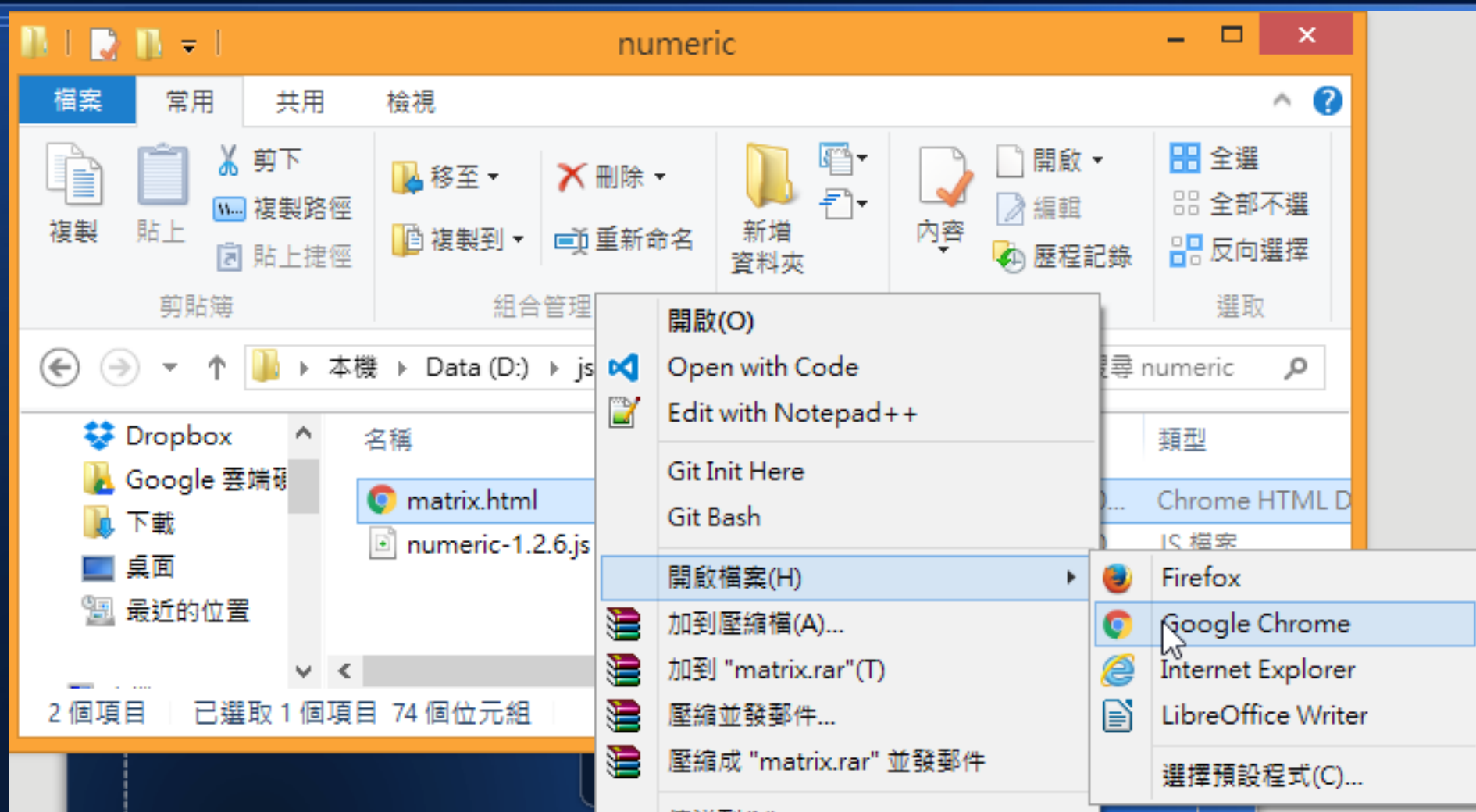
```
1 <html>
2 <body>
3 <script src="numeric-1.2.6.js"></script>
4 </body>
5 </html>
```

The third line, `<script src="numeric-1.2.6.js"></script>`, is highlighted in blue. A red arrow points from the title '記得加上 script 引用' to this line. The status bar at the bottom shows 'length: 74', 'Ln: 3', 'Col: 41', 'Sel: 0 | 0', 'Dos\Windows', 'UTF-8 w/o BOM', and 'INS'.

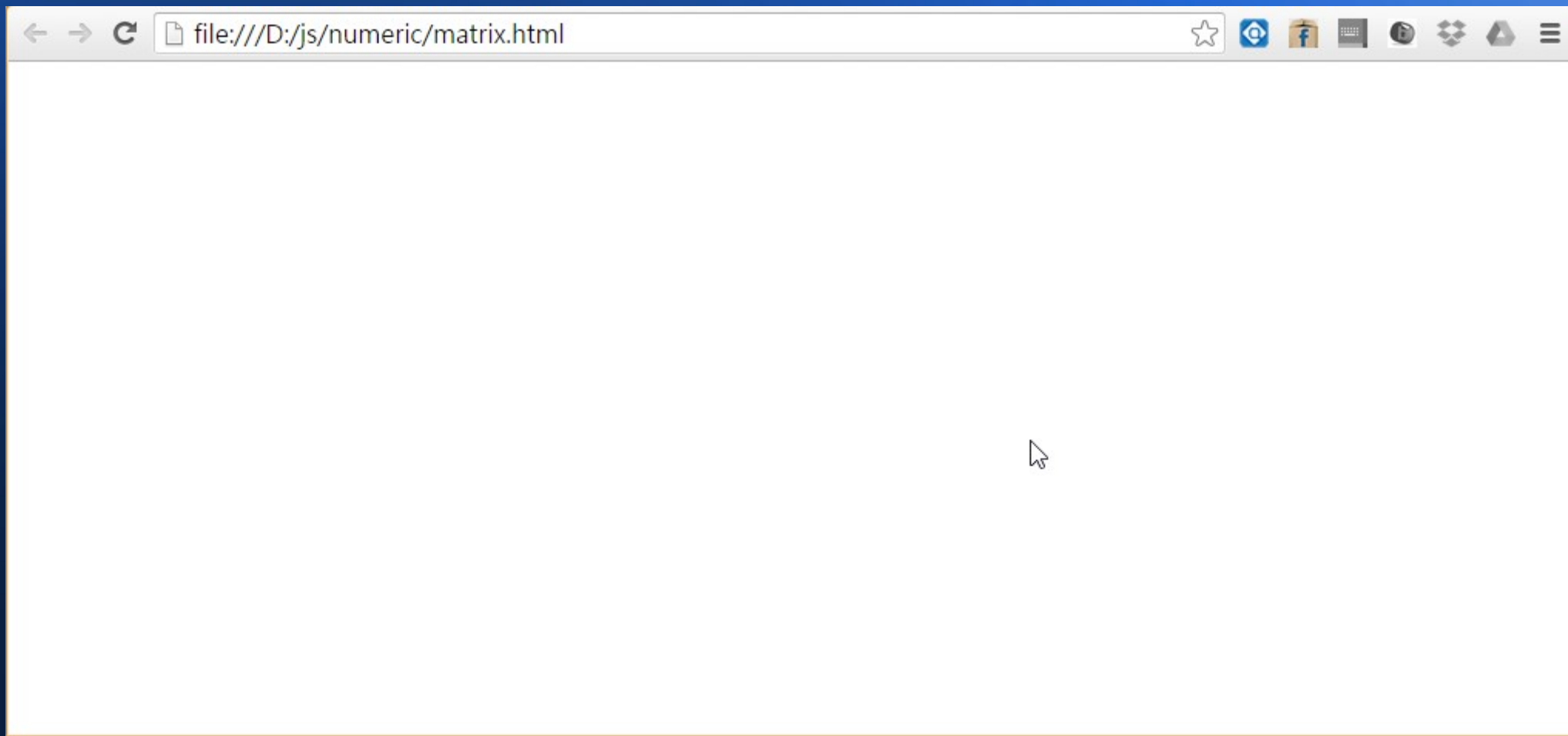
然後就可以存檔



接著用 Chrome 瀏覽器打開 html 檔



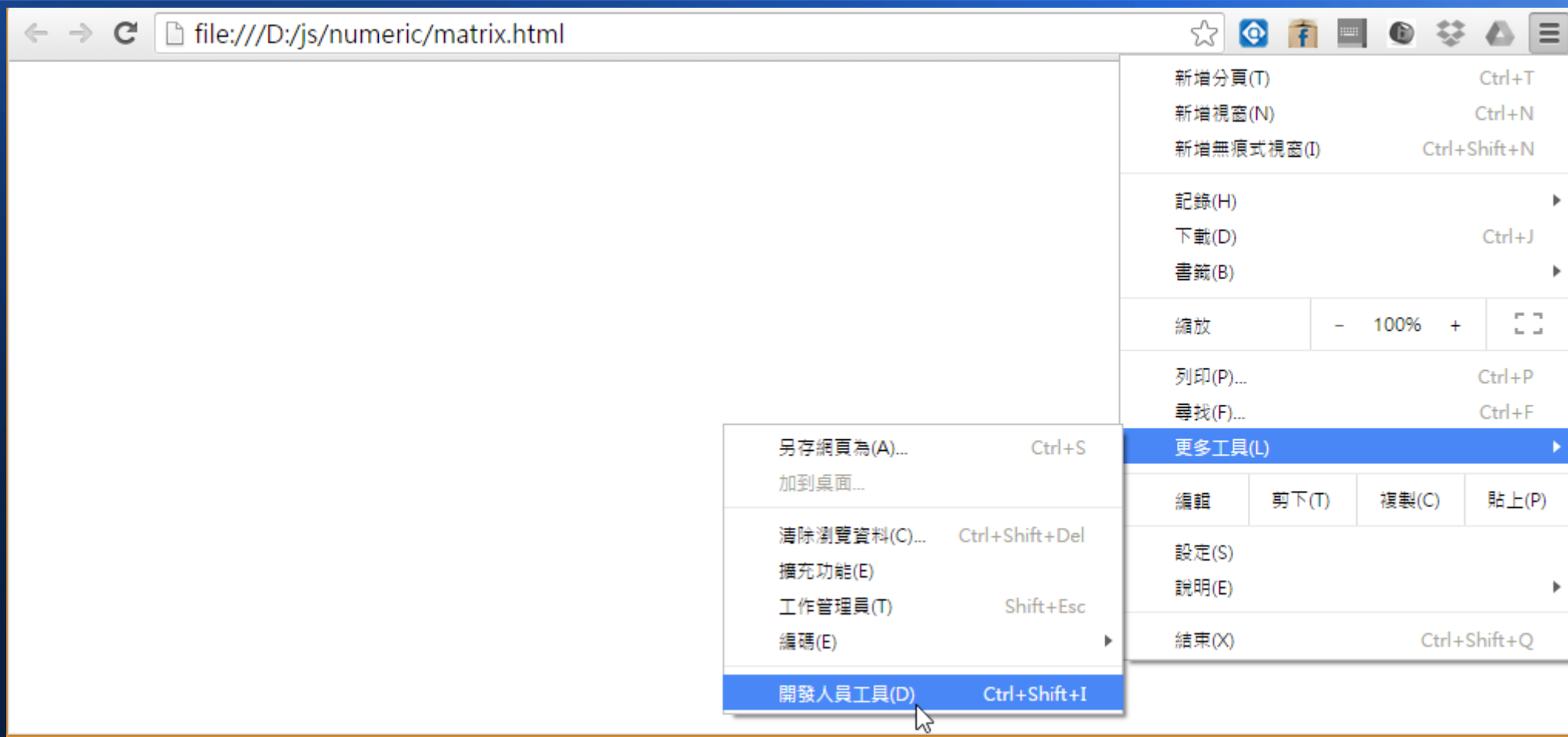
您會看到空空如也！



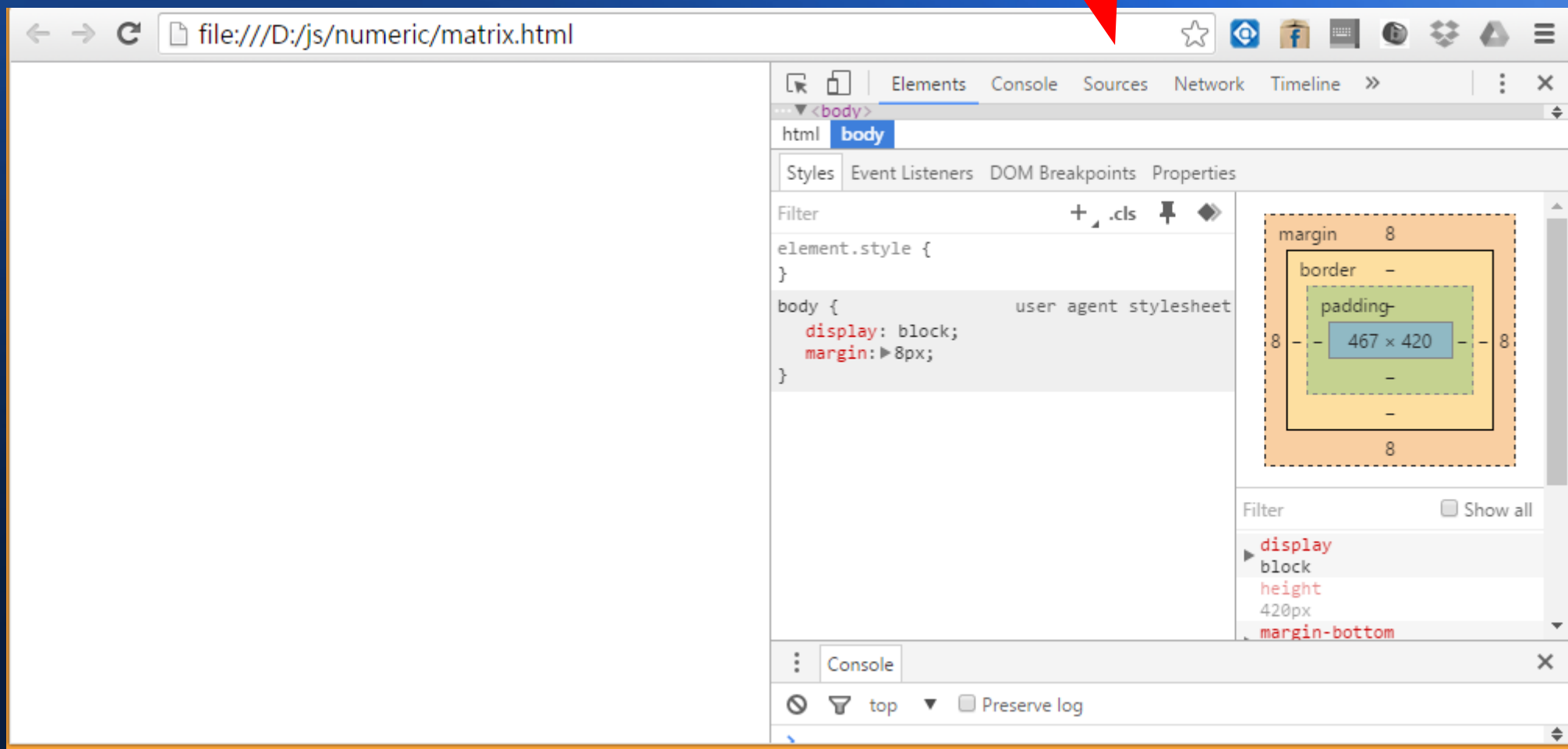
但是沒關係

- 我們有密技！

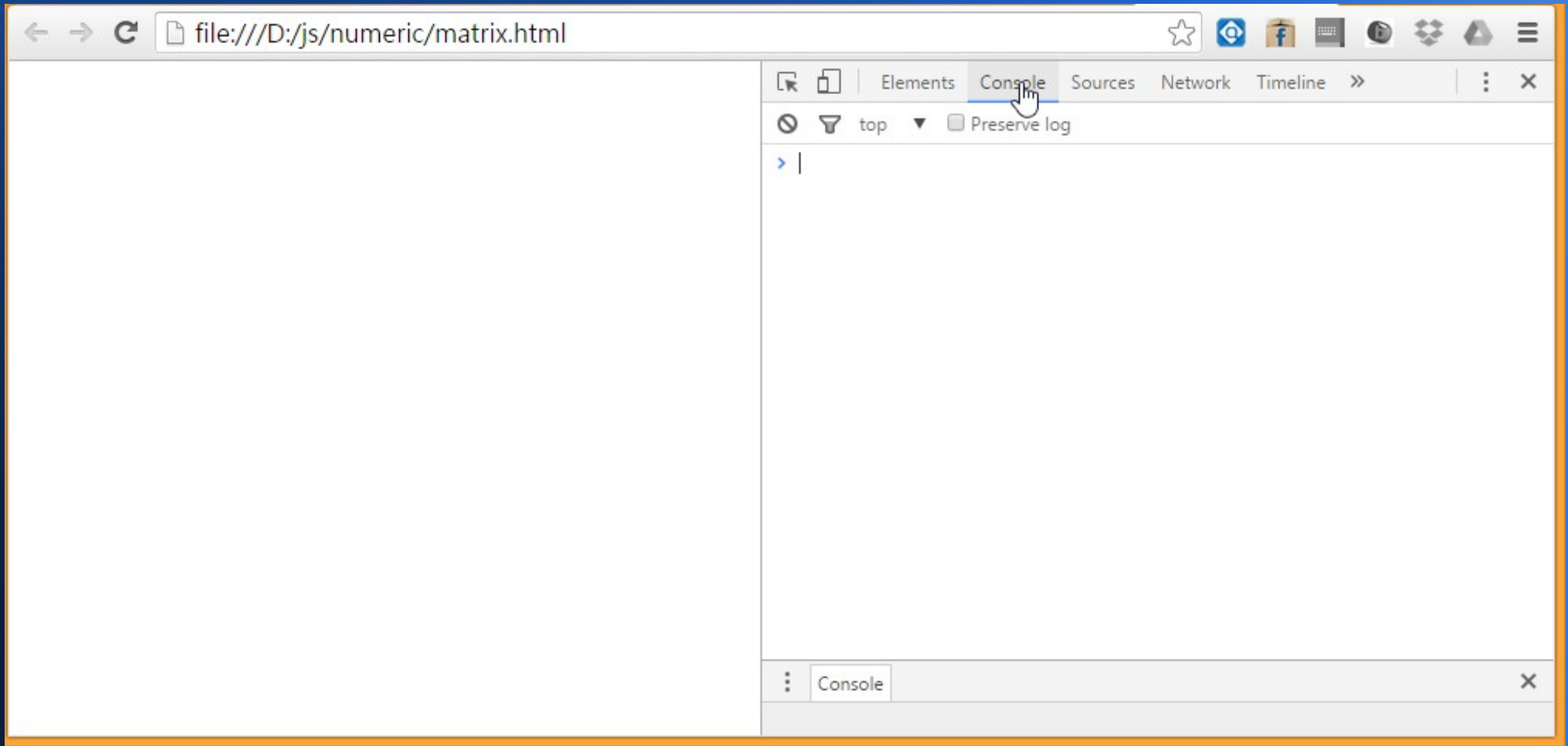
請選《更多工具 / 開發人員工具》



這時您會看到有個窗框跑出來

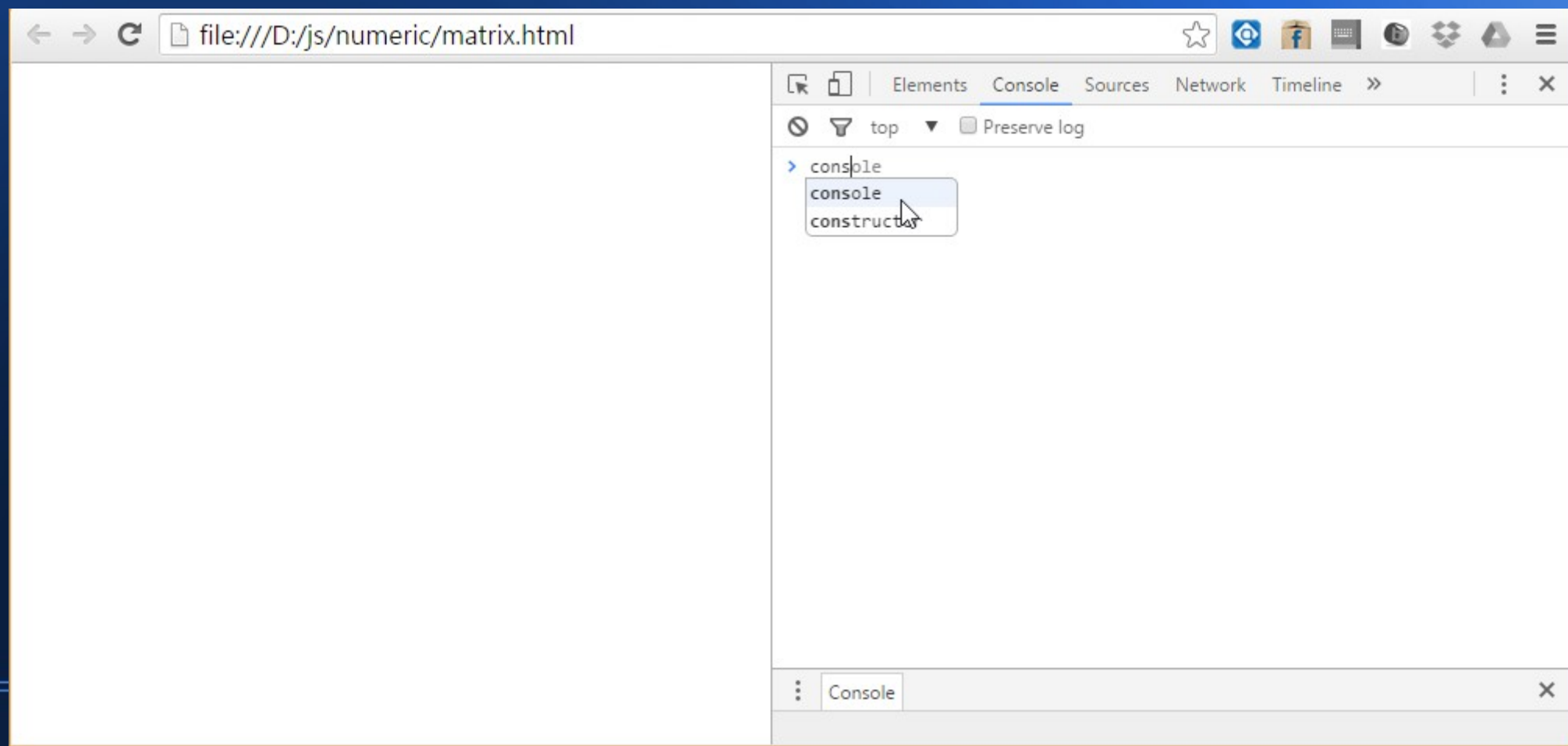


此時您可以選取 console

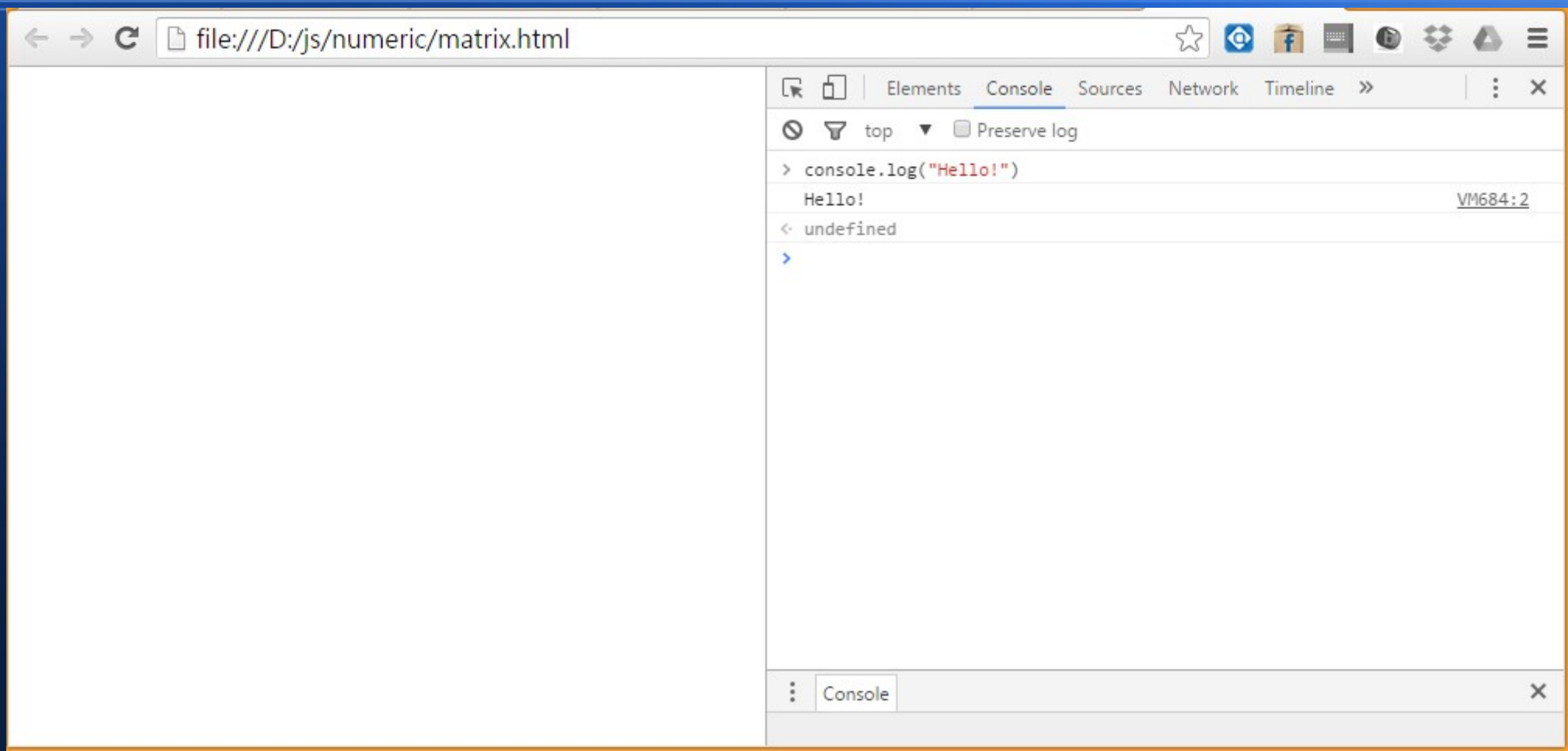


該 console 可以輸入指令

- 而且有很好的提示！

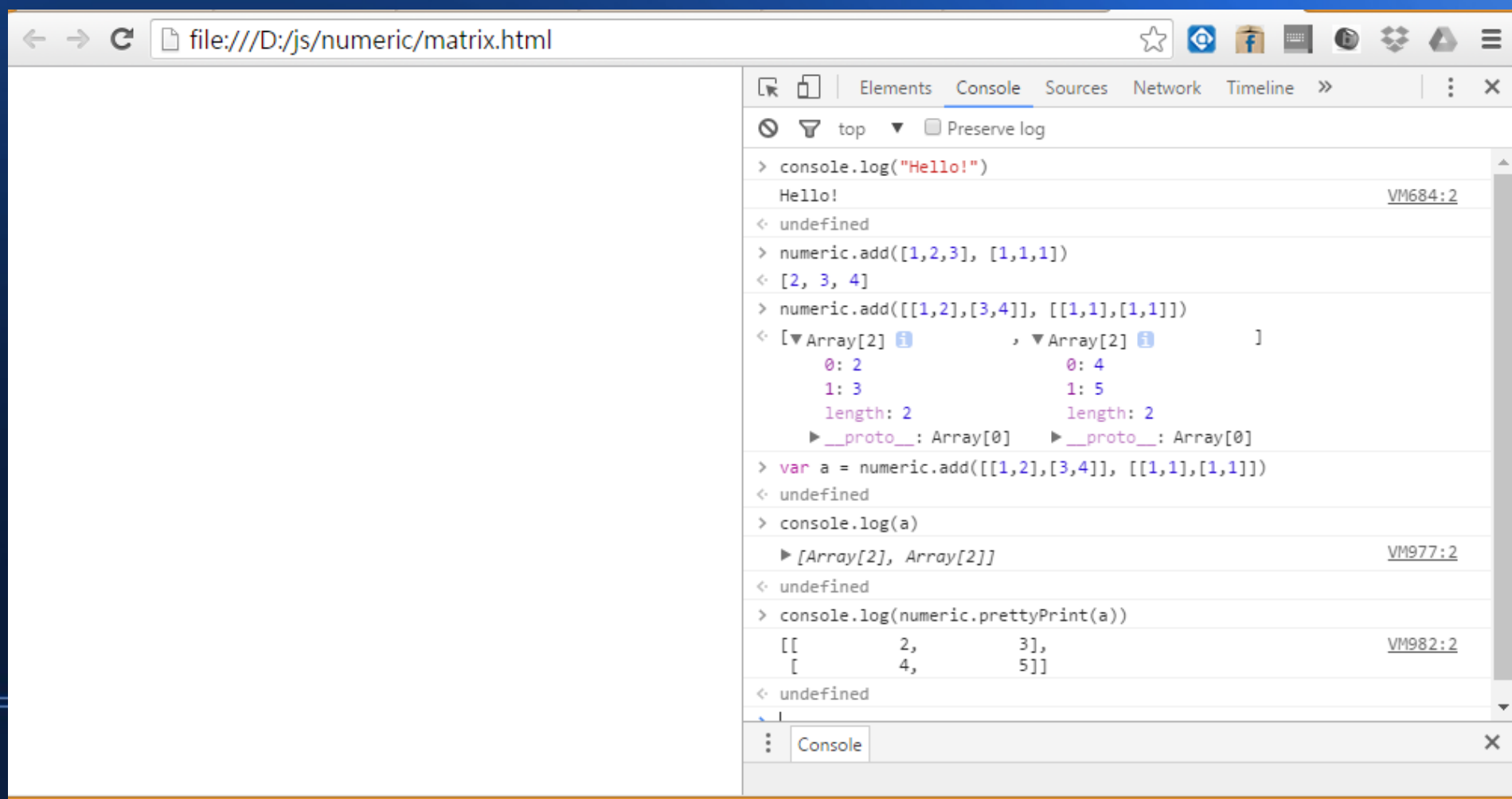


您可以把它當交談式程式環境來用



而您剛剛所引入的 .js 套件

- 也可以在 console 中呼叫並使用

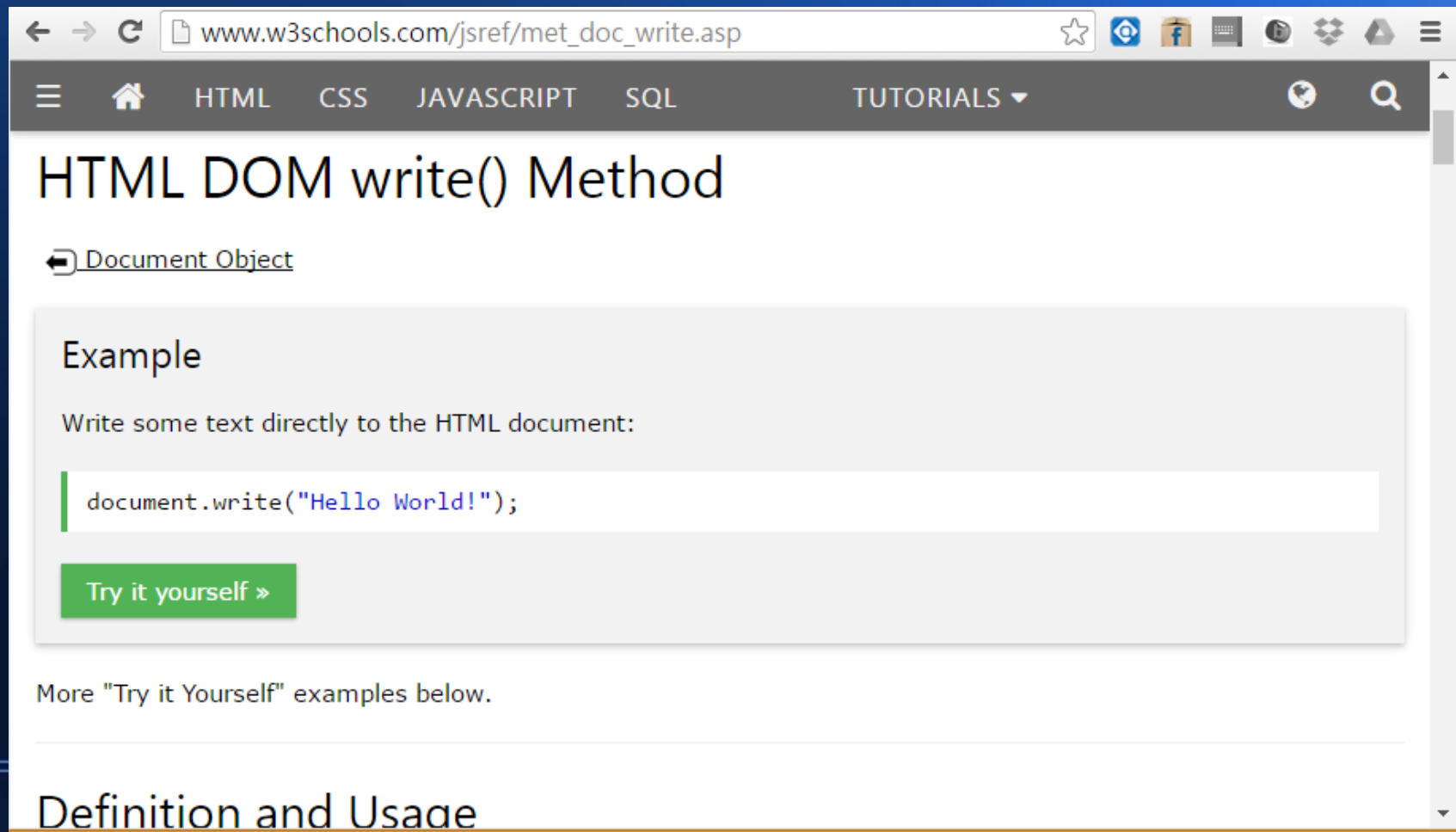


```
file:///D:/js/numeric/matrix.html

> console.log("Hello!")
Hello!
VM684:2
< undefined
> numeric.add([1,2,3], [1,1,1])
< [2, 3, 4]
> numeric.add([[1,2],[3,4]], [[1,1],[1,1]])
< [Array(2), Array(2)]
  0: 2      0: 4
  1: 3      1: 5
  length: 2 length: 2
  __proto__: Array[0]
> var a = numeric.add([[1,2],[3,4]], [[1,1],[1,1]])
< undefined
> console.log(a)
▶ [Array(2), Array(2)]
VM977:2
< undefined
> console.log(numeric.prettyPrint(a))
[[      2,      3],
 [      4,      5]]
VM982:2
< undefined
```

甚至如果你把結果輸出到網頁中

- 例如用 `document.write(...)`



The screenshot shows a web browser window with the address bar displaying `www.w3schools.com/jsref/met_doc_write.asp`. The page title is "HTML DOM write() Method". Below the title, there is a link to "Document Object". The main content area is titled "Example" and contains the text "Write some text directly to the HTML document:". Below this text is a code block containing the JavaScript code `document.write("Hello World!");`. A green button labeled "Try it yourself »" is positioned below the code block. At the bottom of the page, there is a section titled "Definition and Usage".

← → ↻ `www.w3schools.com/jsref/met_doc_write.asp` ☆

≡ 🏠 HTML CSS JAVASCRIPT SQL TUTORIALS 🔍

HTML DOM write() Method

🔗 [Document Object](#)

Example

Write some text directly to the HTML document:

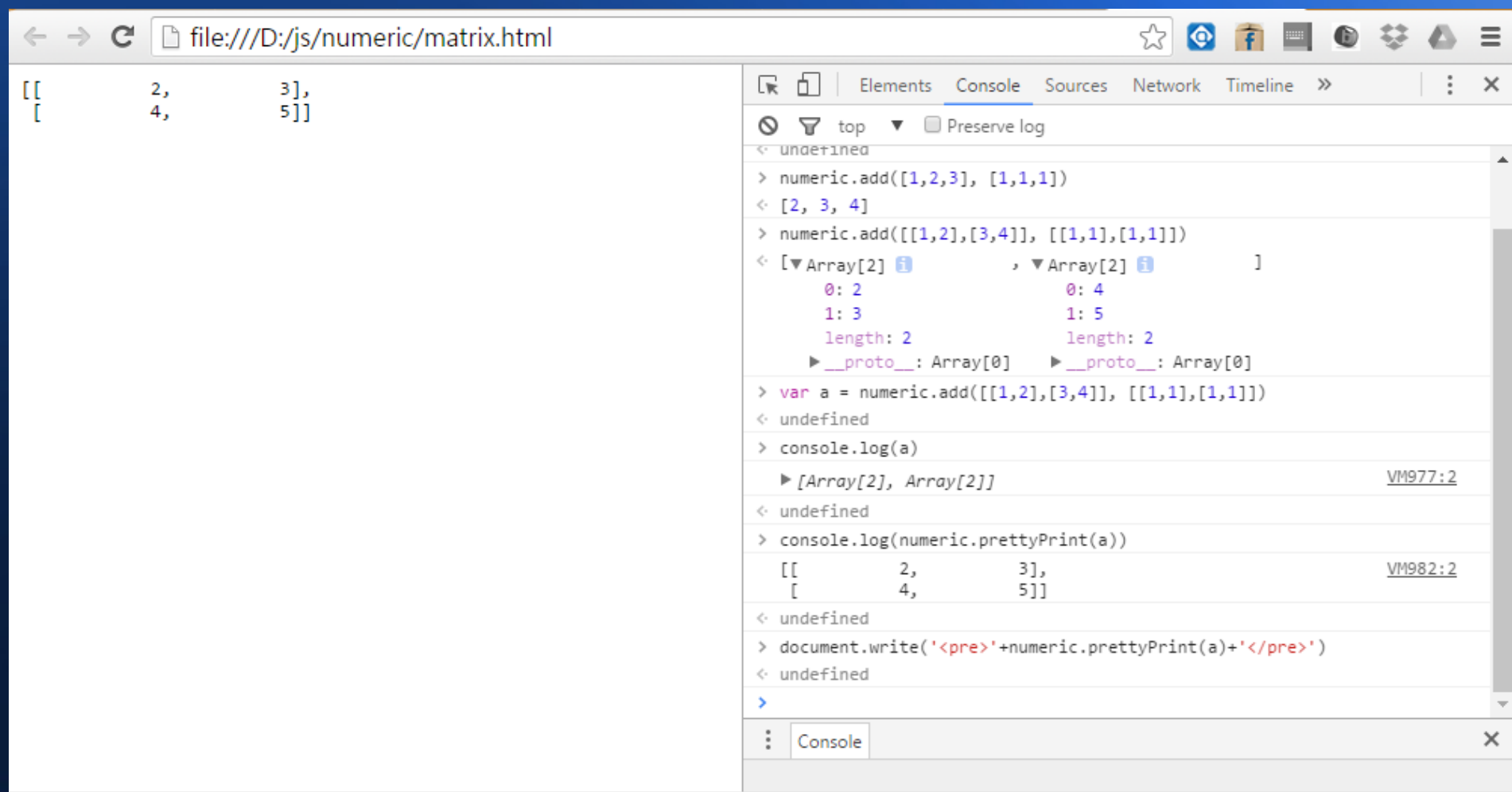
```
document.write("Hello World!");
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

那麼也會看到結果



The screenshot shows a web browser window with the address bar displaying `file:///D:/js/numeric/matrix.html`. The page content on the left shows a JavaScript array: `[[2, 3], [4, 5]]`. The right-hand side of the image shows the browser's developer console with the 'Console' tab selected. The console log shows the following sequence of operations and results:

- `< undefined`
- `> numeric.add([1,2,3], [1,1,1])`
`< [2, 3, 4]`
- `> numeric.add([[1,2],[3,4]], [[1,1],[1,1]])`
`< [Array(2), Array(2)]`
The console shows two array objects side-by-side:
 - `0: 2`, `1: 3`, `length: 2`, `__proto__: Array[0]`
 - `0: 4`, `1: 5`, `length: 2`, `__proto__: Array[0]`
- `> var a = numeric.add([[1,2],[3,4]], [[1,1],[1,1]])`
`< undefined`
- `> console.log(a)`
`> [Array(2), Array(2)]` (labeled `VM977:2`)
- `< undefined`
- `> console.log(numeric.prettyPrint(a))`
`< [[2, 3], [4, 5]]` (labeled `VM982:2`)
- `< undefined`
- `> document.write('<pre>'+numeric.prettyPrint(a)+'</pre>')`
`< undefined`
- `>`

The bottom of the console shows a 'Console' tab with a close button.

這個 Chrome 瀏覽器的開發人員工具

- 就是《前端開發者》，也就是 HTML+CSS+JavaScript 的網頁程式設計者最常用的工具了！

當然

- 您不只可以用 Chrome, 也可以用
火狐的 Firebug
- 當然、您也可以堅持用 IE6 來開發，我是不會阻止你的！

當然、前端開發者

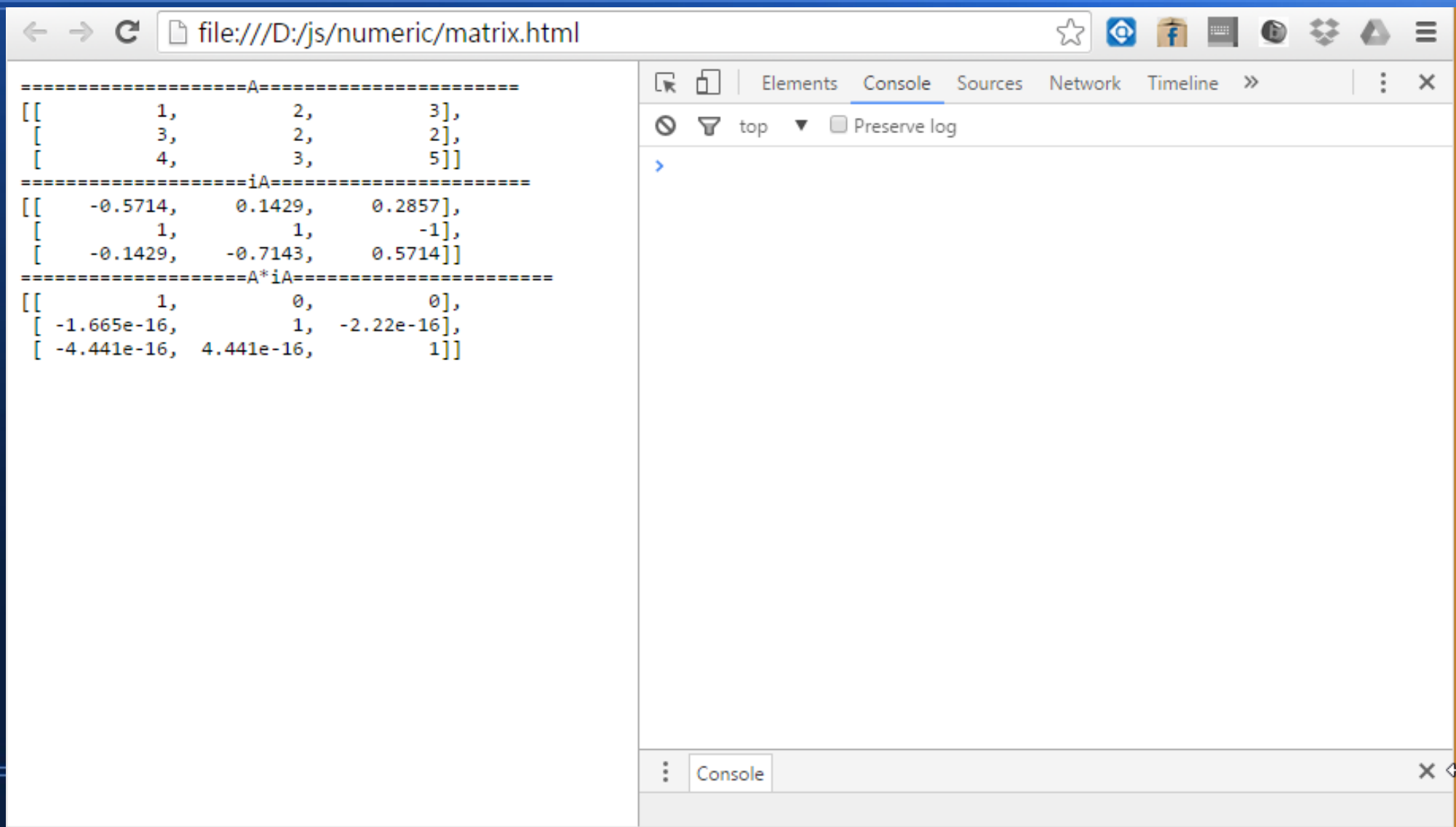
- 要使用該套件，可以直接將
程式碼寫在網頁內！

像是這樣

```
1 <html>
2 <body>
3   <script src="numeric-1.2.6.js"></script>
4   <script>
5     function print(name, M) {
6       document.write("===== "+name+"=====\\n");
7       document.write(numeric.prettyPrint(M)+"\\n");
8     }
9
10    document.write('<pre>');
11
12    var A = [[1,2,3],[3,2,2],[4,3,5]];
13    print("A", A);
14
15    var iA = numeric.inv(A);
16    print("iA", iA);
17
18    var AiA = numeric.dot(A, iA);
19    print("A*iA", AiA);
20
21    document.write('</pre>');
22  </script>
23 </body>
24 </html>
```


然後在開啟該網頁時

- 這些程式就會被執行



The screenshot shows a web browser window with the address bar displaying `file:///D:/js/numeric/matrix.html`. The browser's developer tools are open, showing the Console tab. The console contains the following output:

```
=====A=====
[[      1,      2,      3],
 [      3,      2,      2],
 [      4,      3,      5]]
=====iA=====
[[ -0.5714,  0.1429,  0.2857],
 [      1,      1,     -1],
 [ -0.1429, -0.7143,  0.5714]]
=====A*iA=====
[[      1,      0,      0],
 [-1.665e-16,  1, -2.22e-16],
 [-4.441e-16,  4.441e-16,  1]]
```

The browser's toolbar includes icons for back, forward, refresh, and address bar. The developer tools panel has tabs for Elements, Console, Sources, Network, and Timeline. The Console tab is active, showing the output of the script. The output displays three 3x3 matrices: A, iA, and the result of A*iA. The result matrix is an identity matrix with small floating-point errors in the off-diagonal elements.

於是您就可以在網頁中

- 呼叫此套件並將結果顯示呈現在網頁中！

精於此道的人

- 我們會稱他為前端工程師！

當然

- 既然有《前端工程師》
- 必然有《後端工程師》

對於後端開發者而言


- 目前最強大的 JavaScript 平台
- 當然非 node.js 莫屬了！

問題是

- 我該怎麼在後端開發 node.js 程式呢？

關於這個問題

當然得先安裝 node. js



The screenshot shows the Node.js website homepage. At the top is a dark navigation bar with the Node.js logo and links for HOME, ABOUT, DOWNLOADS, DOCS, FOUNDATION, GET INVOLVED, SECURITY, and NEWS. Below the navigation bar, a paragraph describes Node.js as a JavaScript runtime built on Chrome's V8 JavaScript engine, highlighting its event-driven, non-blocking I/O model and the npm package ecosystem. A green banner below this text announces important security releases and urges users to update. The main content area features a heading for Windows (x64) downloads, followed by two green buttons: 'v4.4.1 LTS Recommended For Most Users' and 'v5.9.1 Stable Latest Features'. At the bottom, there are links for 'Other Downloads', 'Changelog', and 'API Docs' for both versions.

← → ↻ <https://nodejs.org/en/> ☆

node
JS

HOME | ABOUT | DOWNLOADS | DOCS | FOUNDATION | GET INVOLVED | SECURITY | NEWS

Node.js® is a JavaScript runtime built on **Chrome's V8 JavaScript engine**. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, **npm**, is the largest ecosystem of open source libraries in the world.

Important **security releases**, please update now!

Download for Windows (x64)

v4.4.1 LTS
Recommended For Most Users

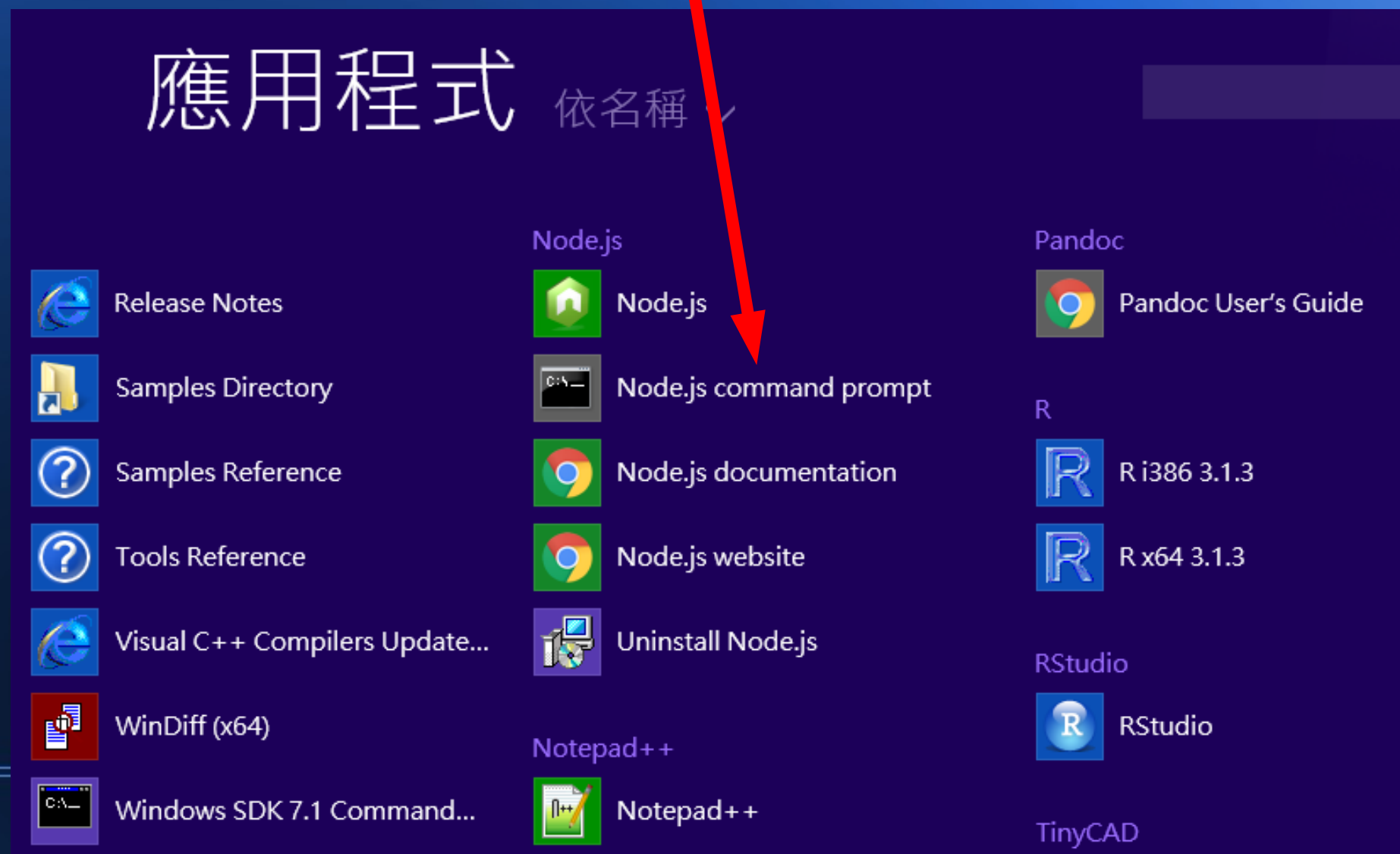
v5.9.1 Stable
Latest Features

Other Downloads | Changelog | API Docs Other Downloads | Changelog | API Docs

安裝的方法很簡單

- 就是下載後執行，然後一直按《下一步》就行了！

如果您在 windows 當中裝完之後
請啟動 node.js command prompt



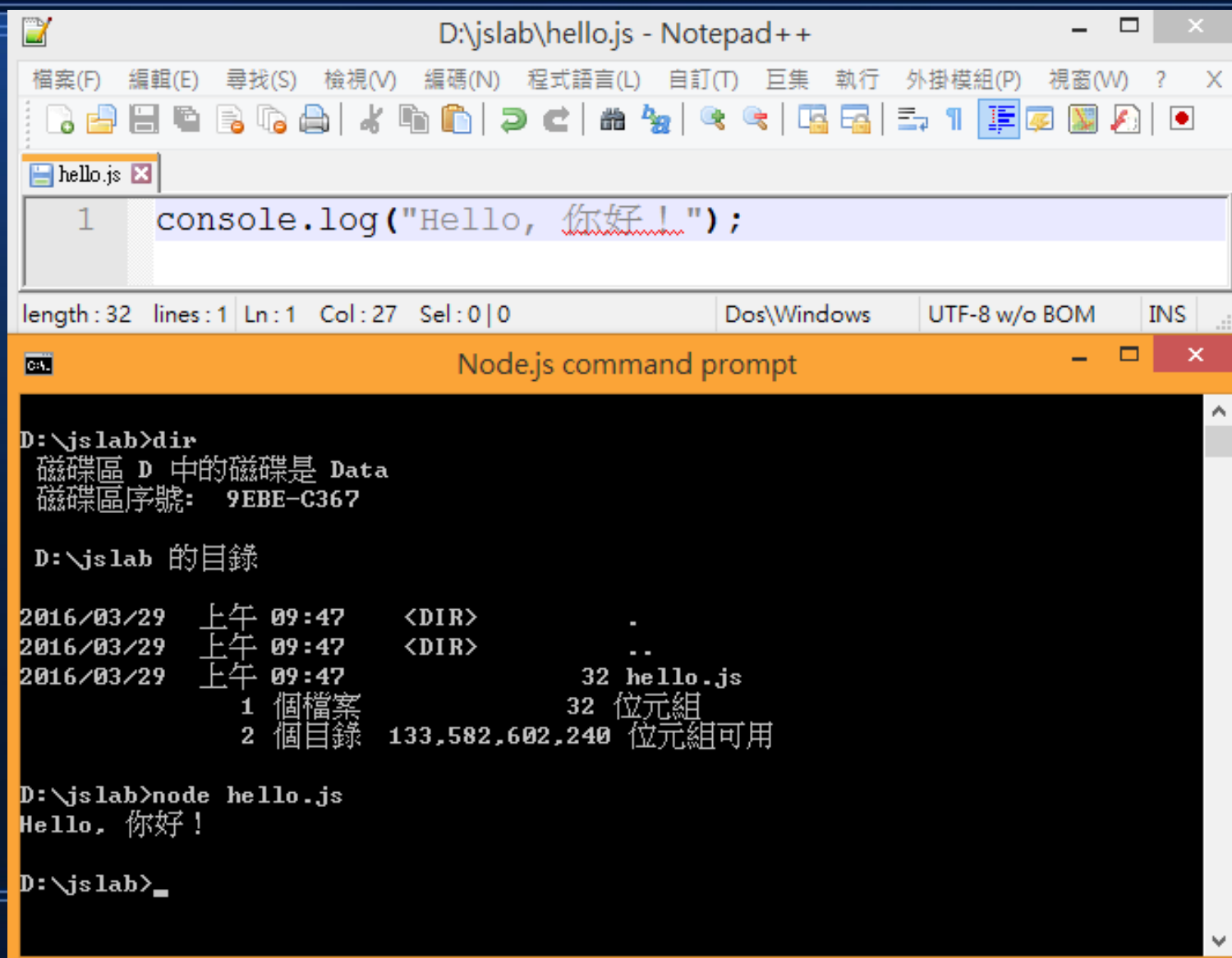
對於使用 MAC 或 Linux 的人

- 則請您啟動《終端機》(Terminal)

如果您還沒用過 node. js

- 可以先試著寫一些小程式
- 像是 hello. js 之類的
- 以便習慣 node. js 的程式環境！

像是這樣



D:\jslab\hello.js - Notepad++

檔案(F) 編輯(E) 尋找(S) 檢視(V) 編碼(N) 程式語言(L) 自訂(T) 巨集 執行 外掛模組(P) 視窗(W) ? X

hello.js

```
1 console.log("Hello, 你好!");
```

length: 32 lines: 1 Ln: 1 Col: 27 Sel: 0|0 Dos\Windows UTF-8 w/o BOM INS

Node.js command prompt

```
D:\jslab>dir
磁碟區 D 中的磁碟是 Data
磁碟區序號: 9EBE-C367

D:\jslab 的目錄
2016/03/29 上午 09:47 <DIR> .
2016/03/29 上午 09:47 <DIR> ..
2016/03/29 上午 09:47 32 hello.js
                  1 個檔案      32 位元組
                  2 個目錄    133,582,602,240 位元組可用

D:\jslab>node hello.js
Hello, 你好!

D:\jslab>
```

如果這些小程序式可以正常執行

- 那麼您就已經開始進入 node.js 的 JavaScript 程式領域了！

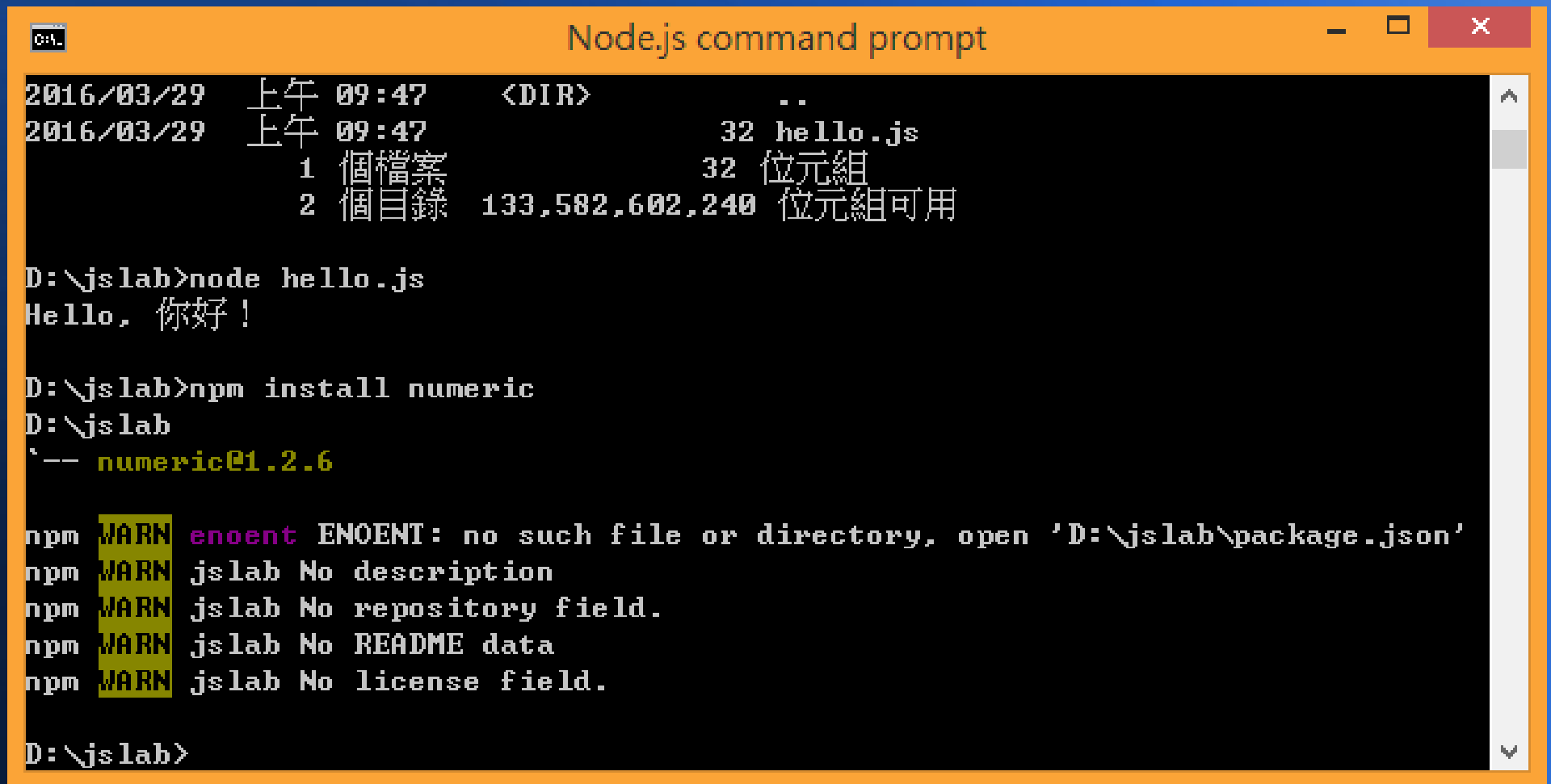
接著您可以開始安裝套件

- 例如用下列指令

`-npm install numeric`

- 安裝剛剛那個 numeric.js 套件

像是這樣



```
Node.js command prompt

2016/03/29 上午 09:47 <DIR>      ..
2016/03/29 上午 09:47          32 hello.js
          1 個檔案          32 位元組
          2 個目錄 133,582,602,240 位元組可用

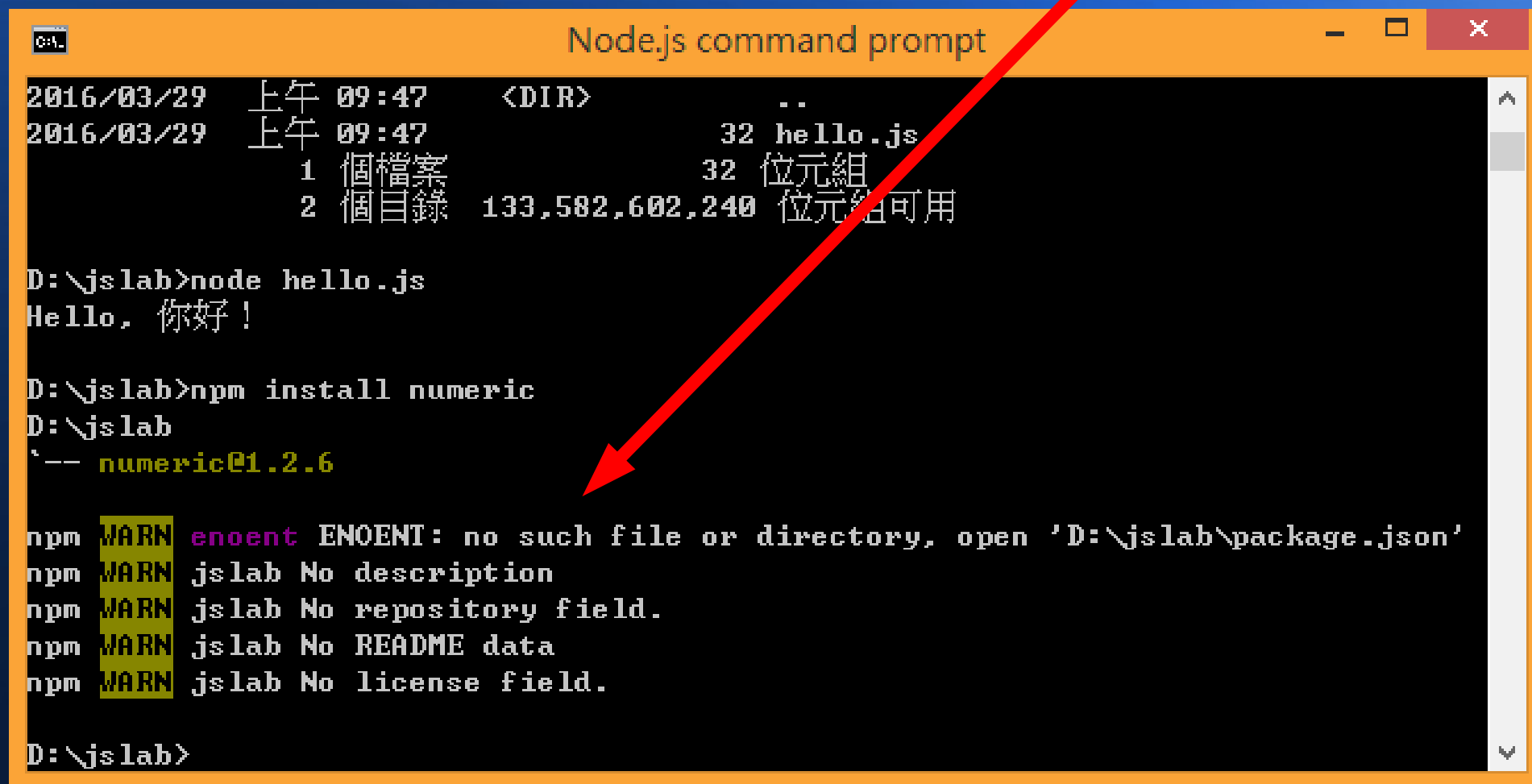
D:\jslab>node hello.js
Hello, 你好!

D:\jslab>npm install numeric
D:\jslab
`-- numeric@1.2.6

npm WARN enoent ENOENT: no such file or directory, open 'D:\jslab\package.json'
npm WARN jslab No description
npm WARN jslab No repository field.
npm WARN jslab No README data
npm WARN jslab No license field.

D:\jslab>
```


奇怪的是、為何有警告訊息？



```
Node.js command prompt

2016/03/29 上午 09:47 <DIR>      ..
2016/03/29 上午 09:47          32 hello.js
          1 個檔案          32 位元組
          2 個目錄 133,582,602,240 位元組可用

D:\jslab>node hello.js
Hello, 你好！

D:\jslab>npm install numeric
D:\jslab
`-- numeric@1.2.6

npm WARN enoent ENOENT: no such file or directory, open 'D:\jslab\package.json'
npm WARN jslab No description
npm WARN jslab No repository field.
npm WARN jslab No README data
npm WARN jslab No license field.

D:\jslab>
```

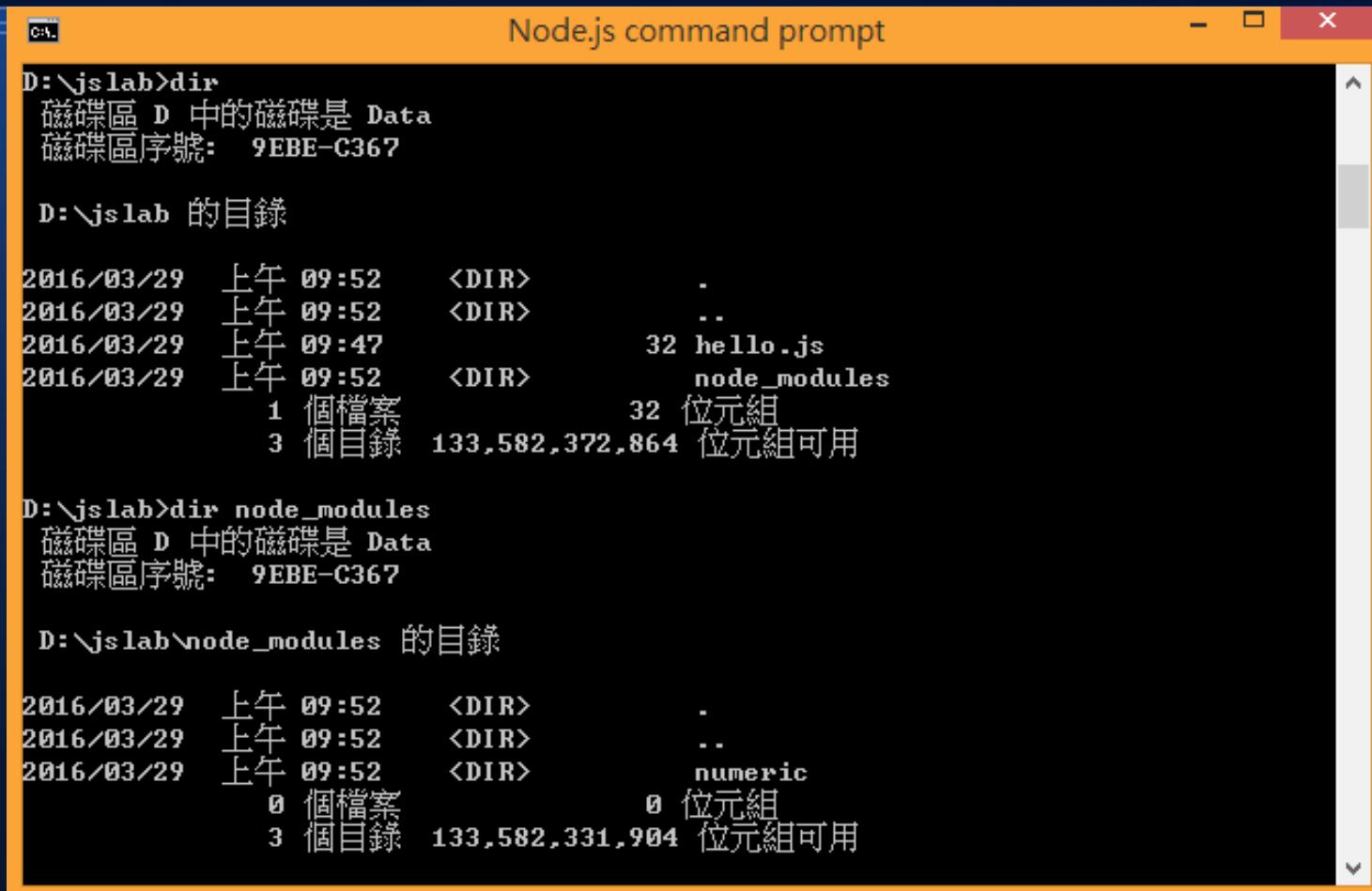
A red arrow points from the title "奇怪的是、為何有警告訊息？" to the warning messages in the terminal output.

請不用擔心

- 那只是告訴你，你沒有定義 package.json 裏的一些欄位
- 因為你根本還沒創建 package.json 這個安裝指引檔案
- 待會我們會說明如何定義 package.json ！

在安裝完 numeric 套件後

- 您會看到出現了 node_modules/numeric 這些子資料夾



```
Node.js command prompt

D:\jslab>dir
磁碟區 D 中的磁碟是 Data
磁碟區字號: 9EBE-C367

D:\jslab 的目錄
2016/03/29 上午 09:52 <DIR> .
2016/03/29 上午 09:52 <DIR> ..
2016/03/29 上午 09:47 32 hello.js
2016/03/29 上午 09:52 <DIR> node_modules
1 個檔案 32 位元組
3 個目錄 133,582,372,864 位元組可用

D:\jslab>dir node_modules
磁碟區 D 中的磁碟是 Data
磁碟區字號: 9EBE-C367

D:\jslab\node_modules 的目錄
2016/03/29 上午 09:52 <DIR> .
2016/03/29 上午 09:52 <DIR> ..
2016/03/29 上午 09:52 <DIR> numeric
0 個檔案 0 位元組
3 個目錄 133,582,331,904 位元組可用
```

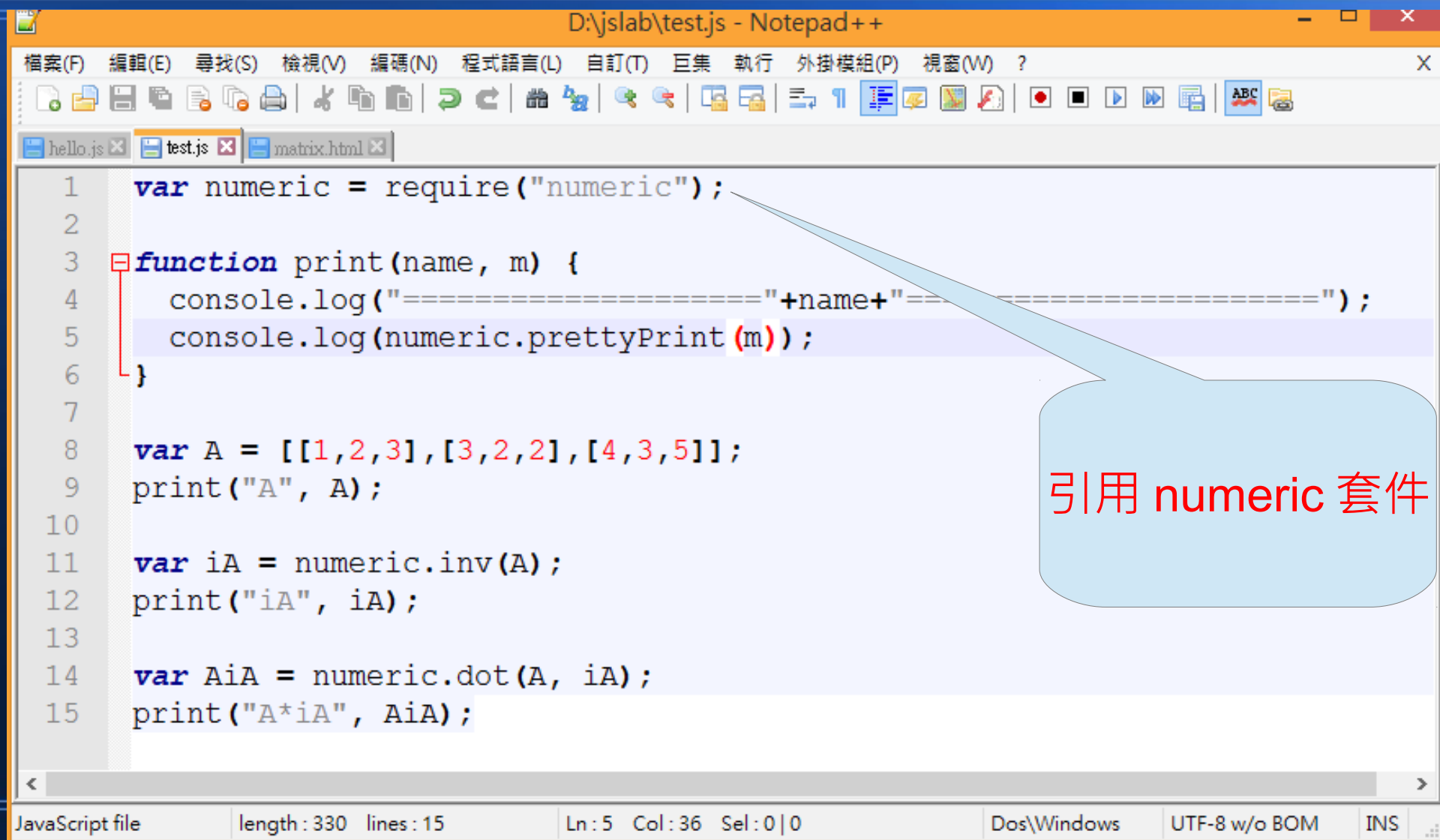
- 那些就是剛剛安裝的套件內容！

現在、讓我們開始在 node.js 裏使用

- 剛剛所安裝的 numeric 套件吧！

我們先寫一個測試程式

- 就叫 test.js 好了！



```
1  var numeric = require("numeric");
2
3  function print(name, m) {
4      console.log("===== "+name+" =====");
5      console.log(numeric.prettyPrint(m));
6  }
7
8  var A = [[1,2,3],[3,2,2],[4,3,5]];
9  print("A", A);
10
11 var iA = numeric.inv(A);
12 print("iA", iA);
13
14 var AiA = numeric.dot(A, iA);
15 print("A*iA", AiA);
```

引用 numeric 套件

JavaScript file length: 330 lines: 15 Ln: 5 Col: 36 Sel: 0 | 0 Dos\Windows UTF-8 w/o BOM INS

然後執行該程式看看

```
Node.js command prompt

D:\jslab>node test.js
=====A=====
[[      1.,      2.,      3],
 [      3.,      2.,      2],
 [      4.,      3.,      5]]
=====iA=====
[[ -0.5714,  0.1429,  0.2857],
 [      1.,      1.,     -1],
 [ -0.1429, -0.7143,  0.5714]]
=====A*iA=====
[[      1.,      0.,      0],
 [ -1.665e-16,  1.,  -2.22e-16],
 [ -4.441e-16,  4.441e-16,  1]]

D:\jslab>
```

如果能正常執行

- 那就代表引用套件成功
- 而且您使用該套件的方法是正確的！

寫到這裡

- 您以為故事講完了嗎！

其實

- 剛好相反！

本文的主題

- 才剛剛開始而已！

因為我們的主題是

用十分鐘瞭解 JavaScript 的模組

《還有關於 npm 套件管理的那些事情》

陳鍾誠

2016 年 3 月 29 日

但是、到目前為止

- 我們都還沒開始

撰寫自己的模組呢？

問題是

- 我們要怎麼樣撰寫自己的模組呢？

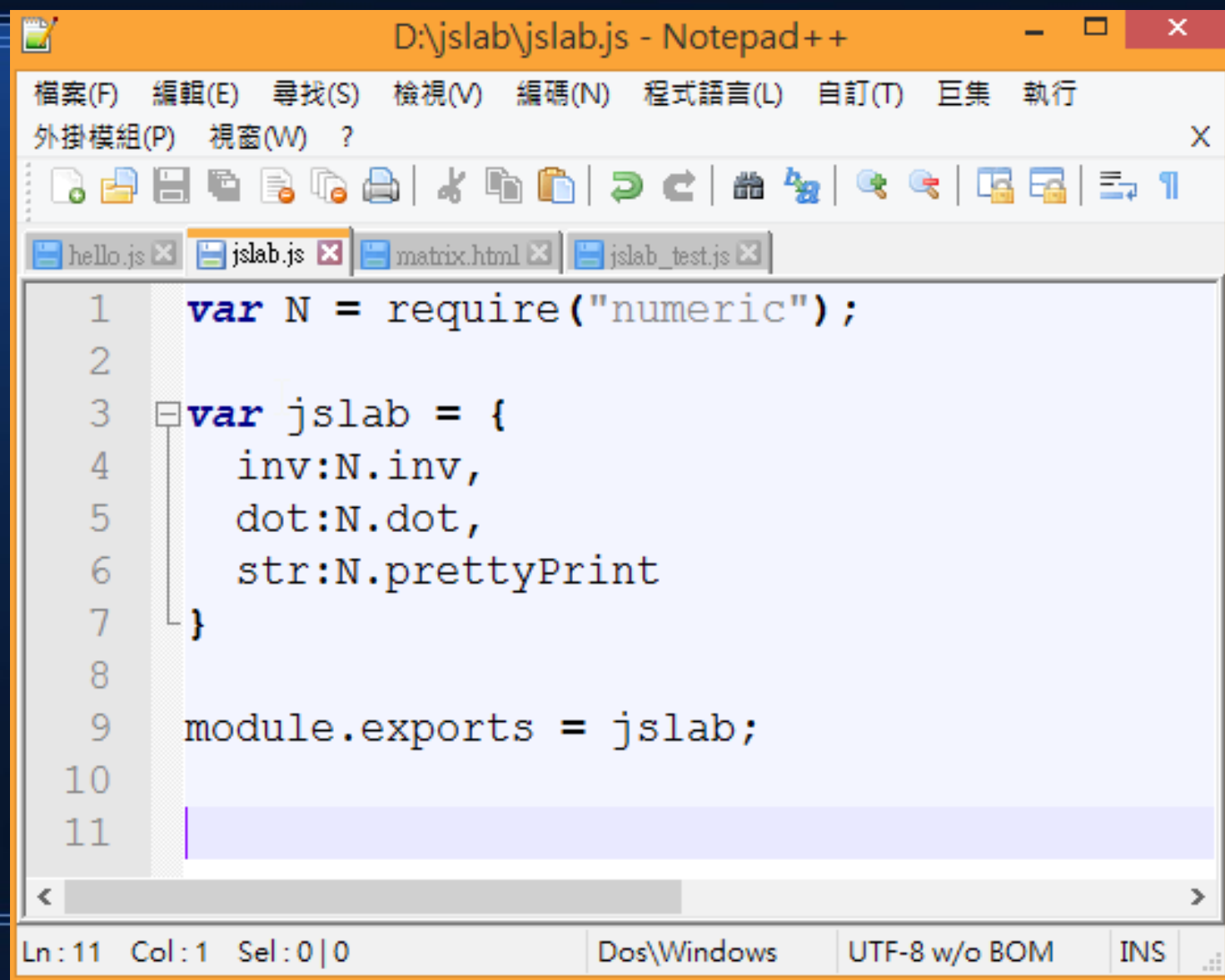
方法不難

- 一樣是寫 .js 程式
- 只是最後要加上
 - `module.exports = <object>`
- 這樣才能給其他程式使用

例如

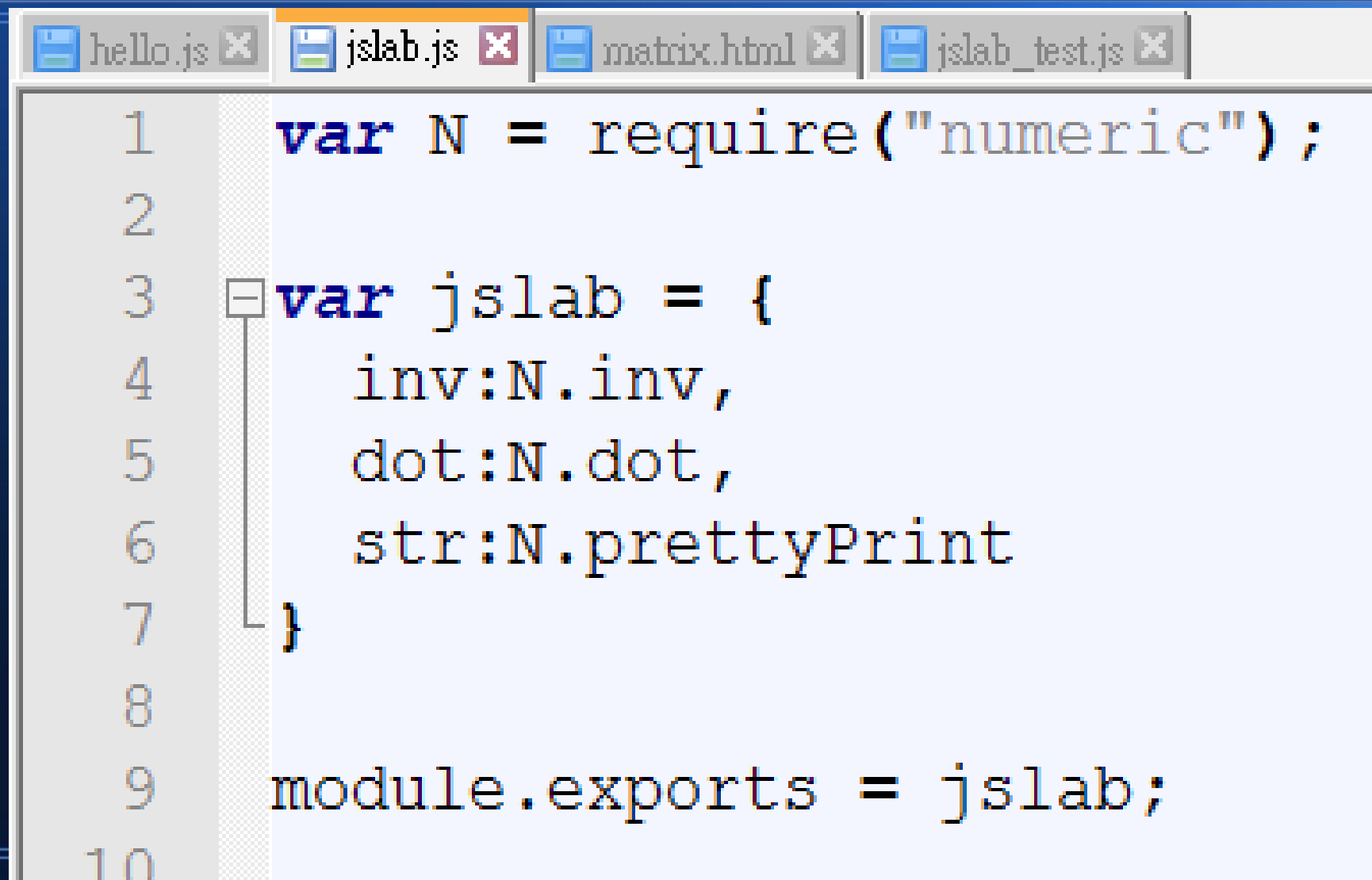
- 我們可以用剛剛那個 `numeric.js` 套件為基礎
- 重新包裝出自己的 `jslab.js` 這個科學計算函式庫。

以下是 jslab.js 的程式碼



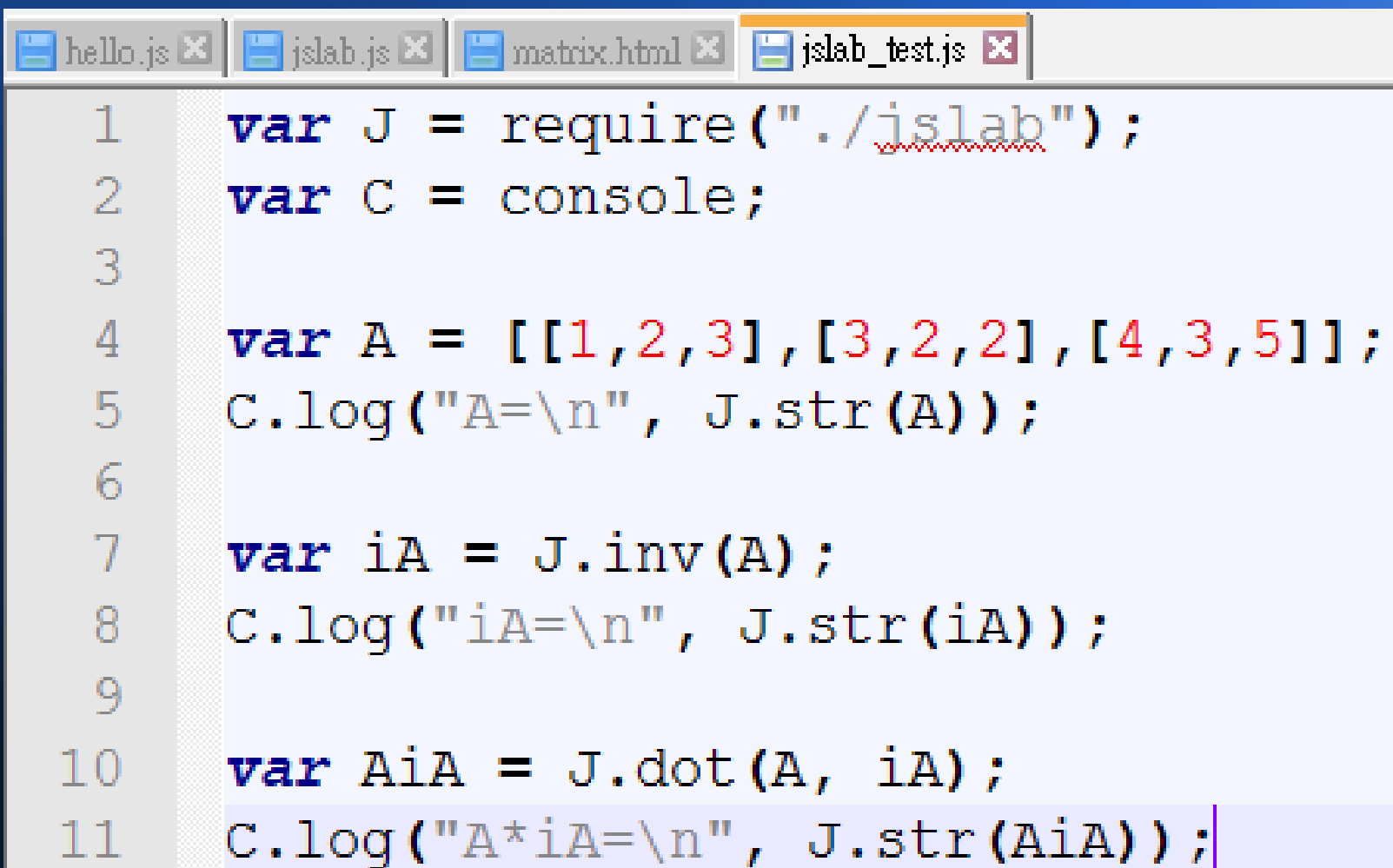
```
1  var N = require("numeric");
2
3  var jslab = {
4    inv:N.inv,
5    dot:N.dot,
6    str:N.prettyPrint
7  }
8
9  module.exports = jslab;
10
11
```


您可以看到、我們只不過把 `numeric` 重新包裝成 `jslab`，然後就用 `module.exports` 匯出了



```
1  var N = require("numeric");
2
3  var jslab = {
4      inv:N.inv,
5      dot:N.dot,
6      str:N.prettyPrint
7  }
8
9  module.exports = jslab;
10
```

接著我們就可以寫一個測試程式
jslab_test 來使用這個 jslab 模組

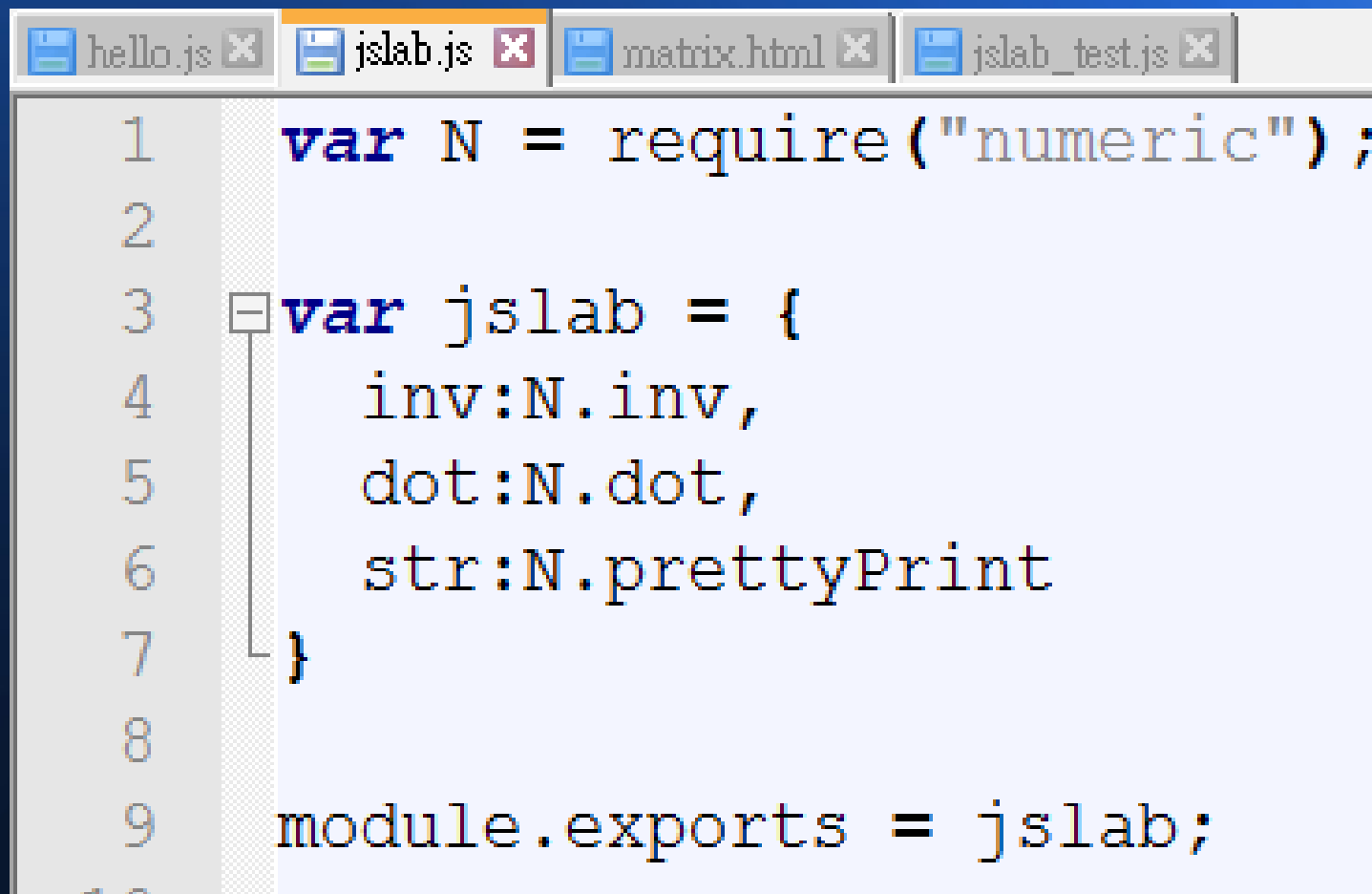


```
1  var J = require("./jslab");
2  var C = console;
3
4  var A = [[1,2,3],[3,2,2],[4,3,5]];
5  C.log("A=\n", J.str(A));
6
7  var iA = J.inv(A);
8  C.log("iA=\n", J.str(iA));
9
10 var AiA = J.dot(A, iA);
11 C.log("A*iA=\n", J.str(AiA));
```

jslab_test 的執行結果如下

```
Node.js command
D:\jslab>node jslab_test
A=
[[      1.,      2.,      3],
 [      3.,      2.,      2],
 [      4.,      3.,      5]]
iA=
[[ -0.5714,  0.1429,  0.2857],
 [      1.,      1.,     -1],
 [ -0.1429, -0.7143,  0.5714]]
A*iA=
[[      1.,      0.,      0],
 [-1.665e-16,  1., -2.22e-16],
 [-4.441e-16,  4.441e-16,  1]]
D:\jslab>
```

換句話說、每個程式
只要用 `module.exports` 匯出物件

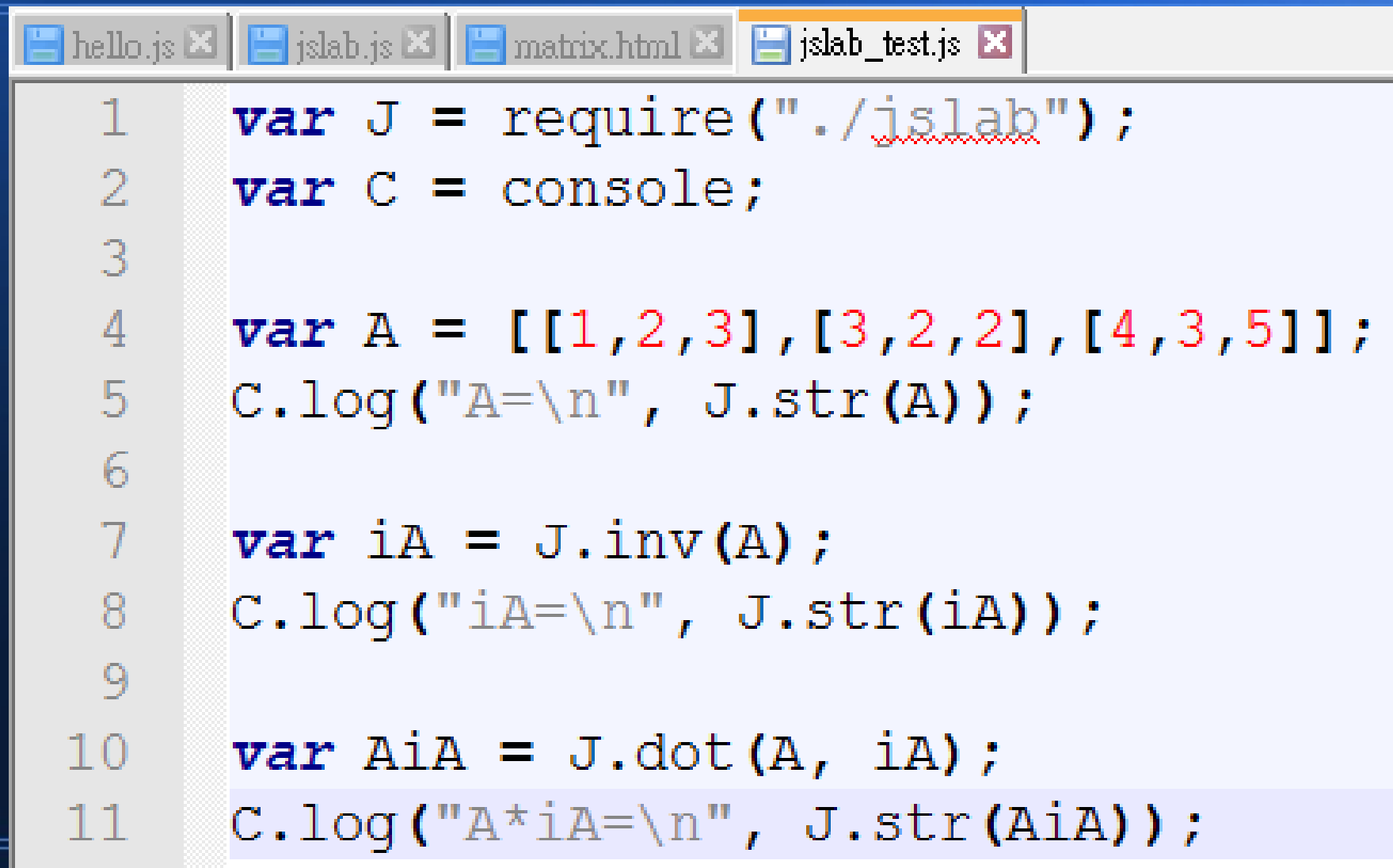


The screenshot shows a code editor with four tabs: `hello.js`, `jslab.js` (active), `matrix.html`, and `jslab_test.js`. The `jslab.js` file contains the following code:

```
1  var N = require("numeric");
2
3  var jslab = {
4      inv:N.inv,
5      dot:N.dot,
6      str:N.prettyPrint
7  }
8
9  module.exports = jslab;
```

A mouse cursor is positioned over the opening curly brace of the `jslab` object on line 3.

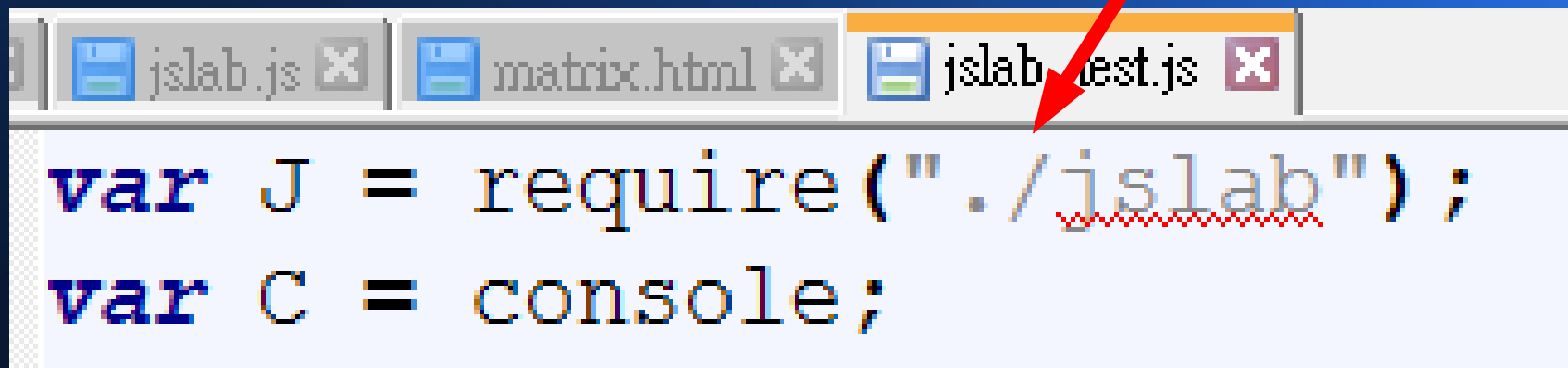
接著就可以用 `require()` 函數 引入該物件來使用



```
1  var J = require("./jslab");
2  var C = console;
3
4  var A = [[1,2,3],[3,2,2],[4,3,5]];
5  C.log("A=\n", J.str(A));
6
7  var iA = J.inv(A);
8  C.log("iA=\n", J.str(iA));
9
10 var AiA = J.dot(A, iA);
11 C.log("A*iA=\n", J.str(AiA));
```

但必須注意的是

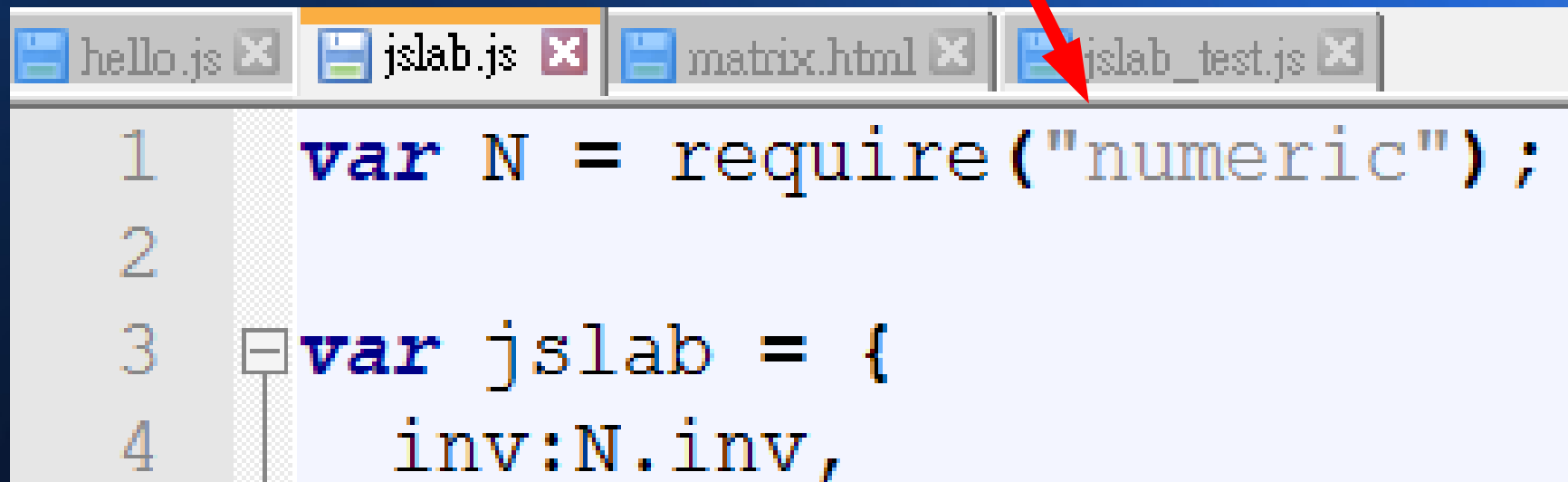
- 引入自己定義的模組，必須要指定路徑，如果兩者放在同一資料夾，那也必須加上 `./` 的符號，告訴 `node.js` 該檔案在同一資料夾底下。



```
var J = require("./jslab");  
var C = console;
```

不過如果是引用

- 系統預設的模組或者 npm 安裝的套件
- 那麼就不用加上 `./`，直接寫模組名稱就可以了。



```
1  var N = require("numeric");
2
3  var jslab = {
4      inv:N.inv,
```

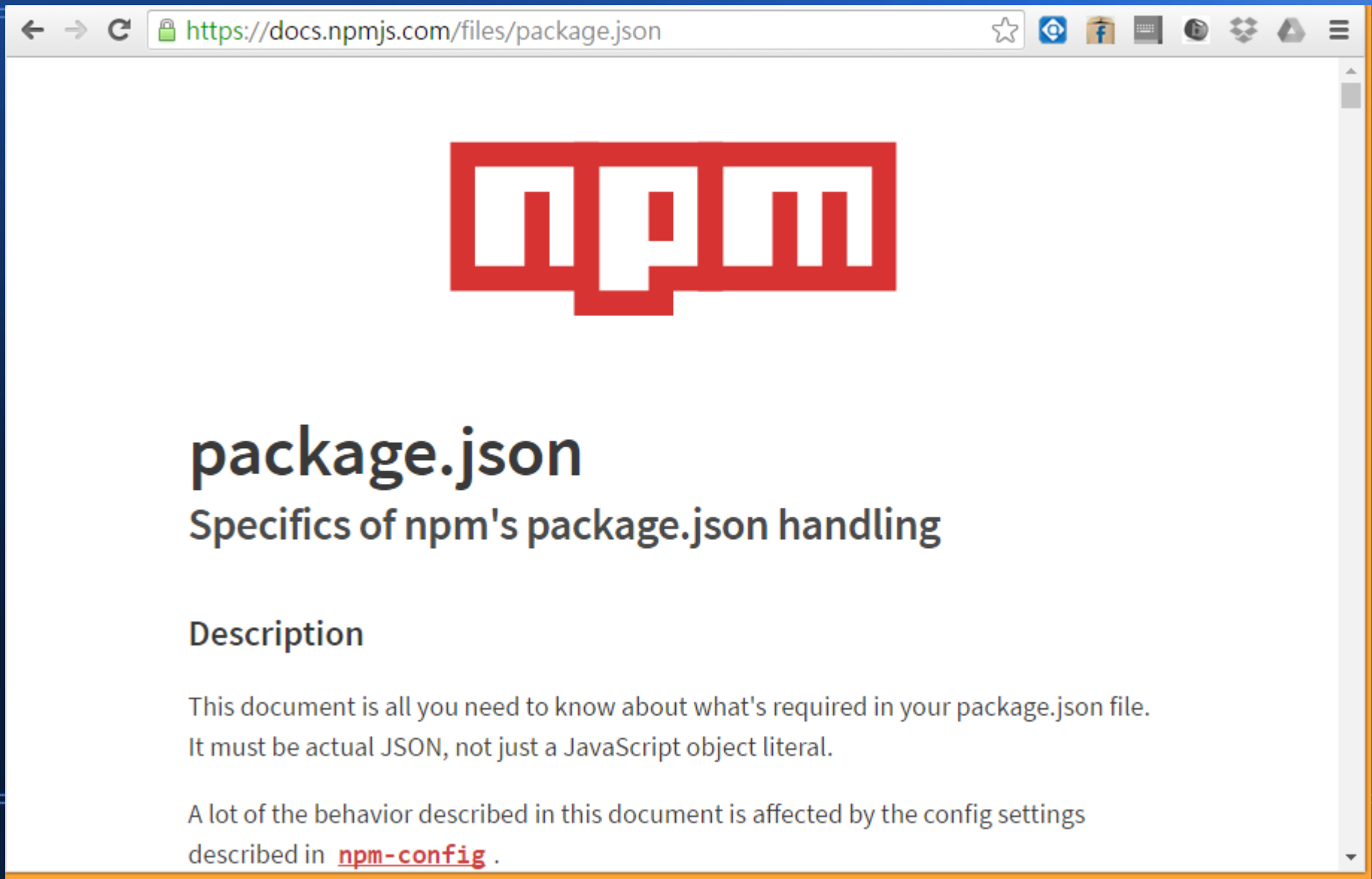
好了、現在我們已經學會了

- 如何自己定義模組了！

但問題是

- 我寫的模組要怎麼發佈給別人使用呢？

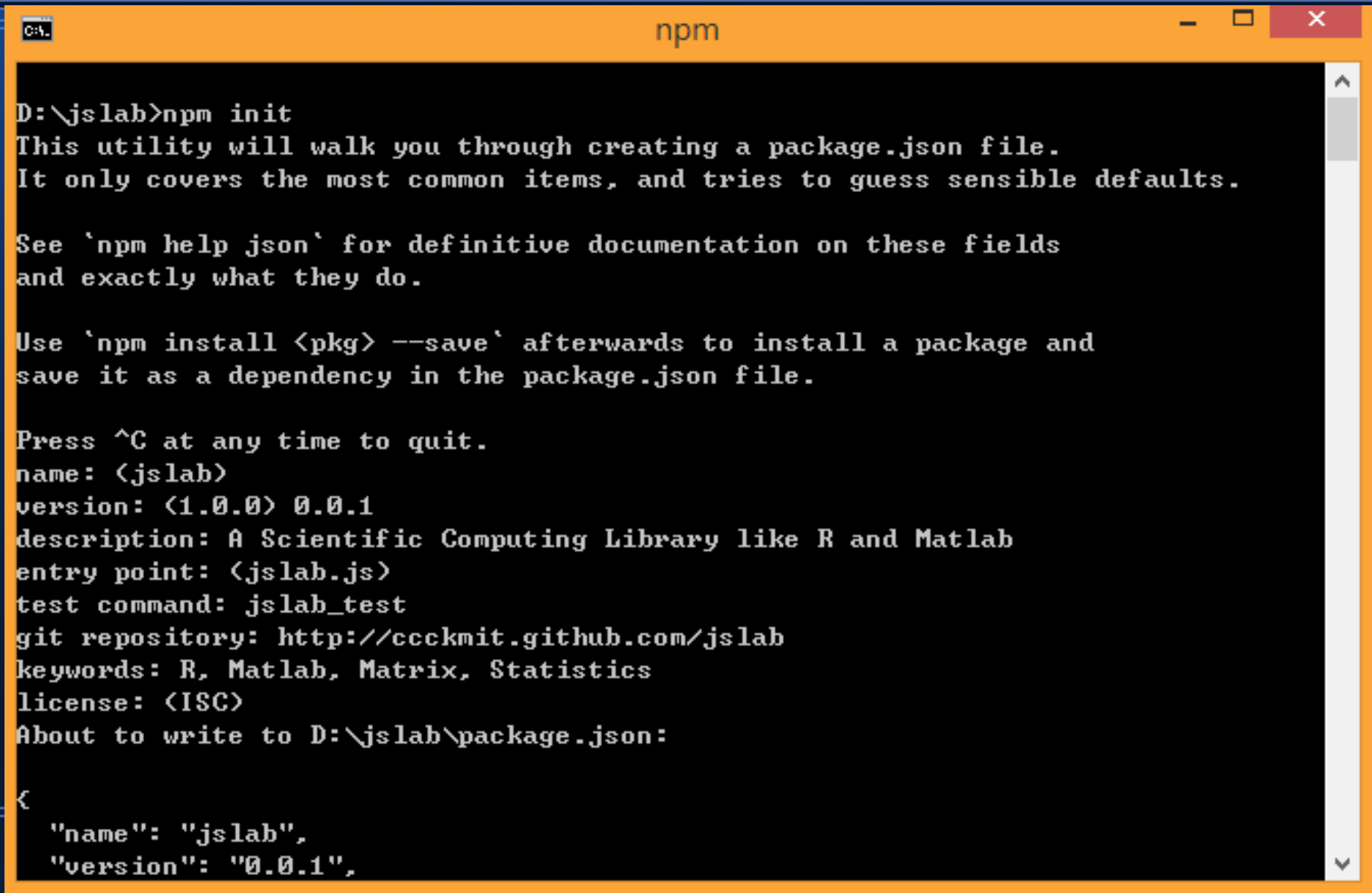
關於這個問題，就必須要請 package.json 登場了



package.json

- 是用來描述《發佈模組》的一個檔案。
- 其中會記載《套件名稱、版本代號、作者、套件網址.....》等資訊。

您可以用 `npm init` 指令 來新建一個 `package.json` 檔案



```
C:\> npm init
D:\jslab> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

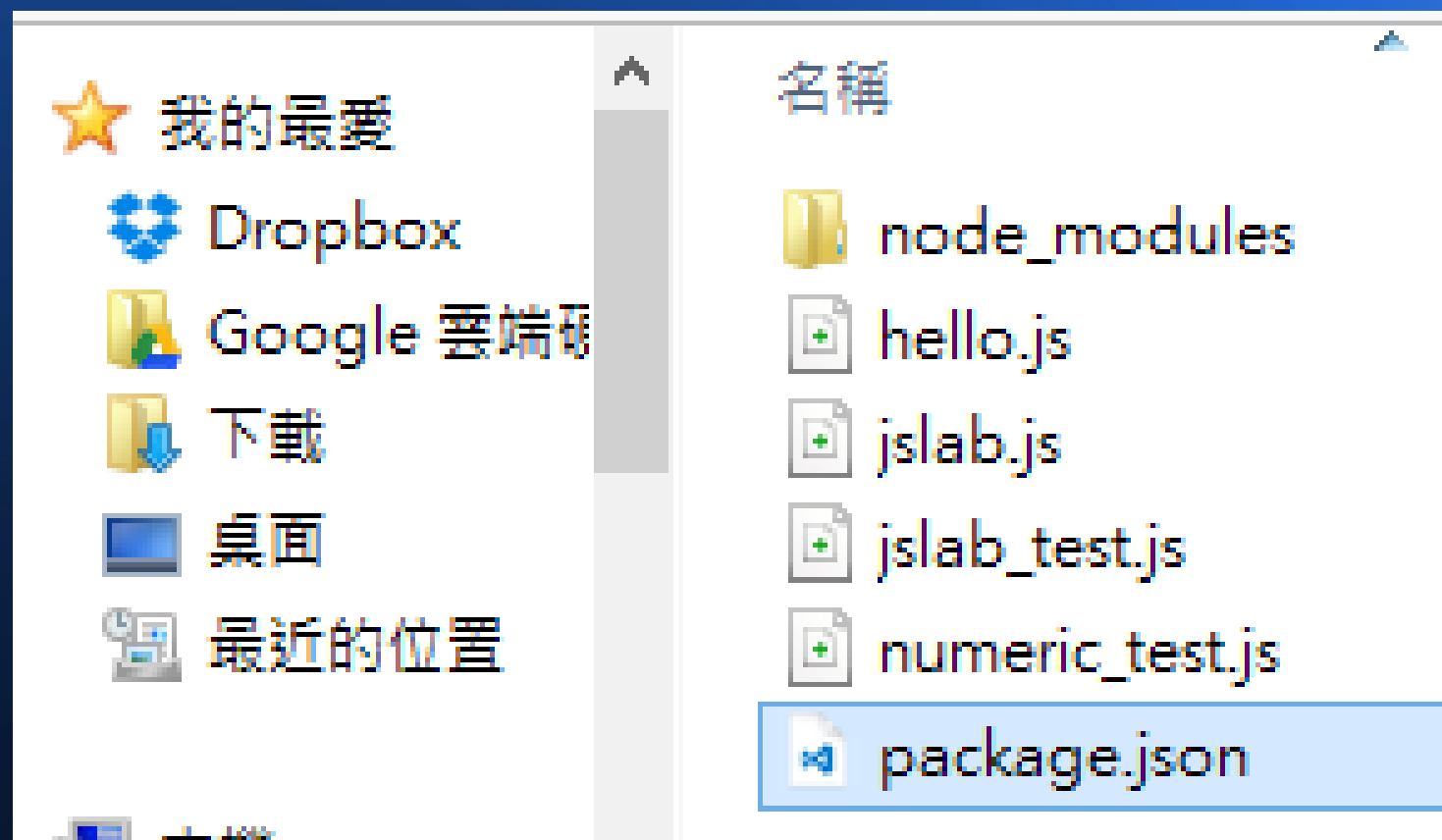
See 'npm help json' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg> --save' afterwards to install a package and
save it as a dependency in the package.json file.

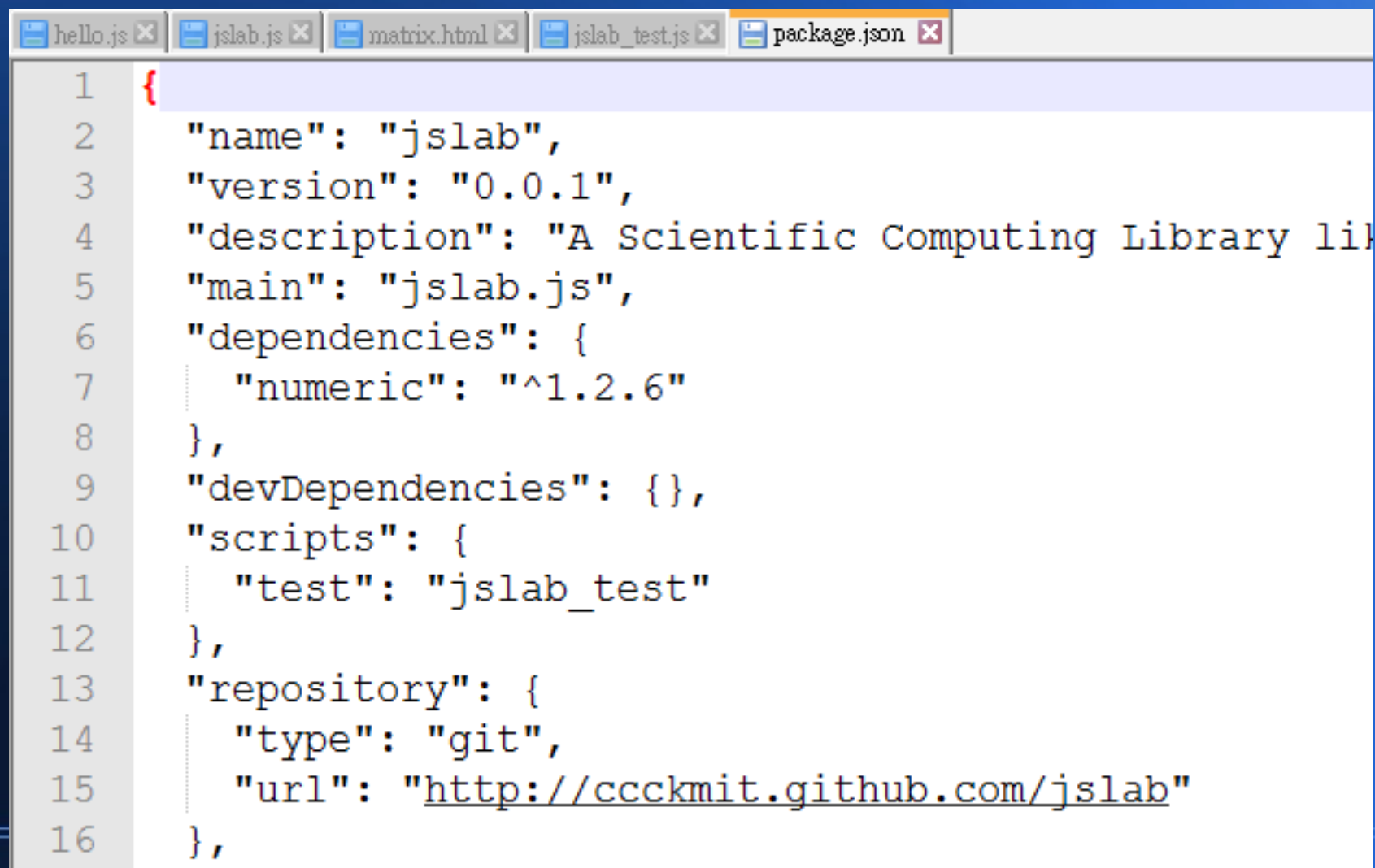
Press ^C at any time to quit.
name: <jslab>
version: <1.0.0> 0.0.1
description: A Scientific Computing Library like R and Matlab
entry point: <jslab.js>
test command: jslab_test
git repository: http://ccckmit.github.com/jslab
keywords: R, Matlab, Matrix, Statistics
license: <ISC>
About to write to D:\jslab\package.json:

{
  "name": "jslab",
  "version": "0.0.1",
```

當您回答完一堆問題後，`npm init` 就會產生 `package.json`



產生的 package.json 內容如下



The image shows a code editor window with five tabs: 'hello.js', 'jslab.js', 'matrix.html', 'jslab_test.js', and 'package.json'. The 'package.json' tab is active, displaying the following JSON content:

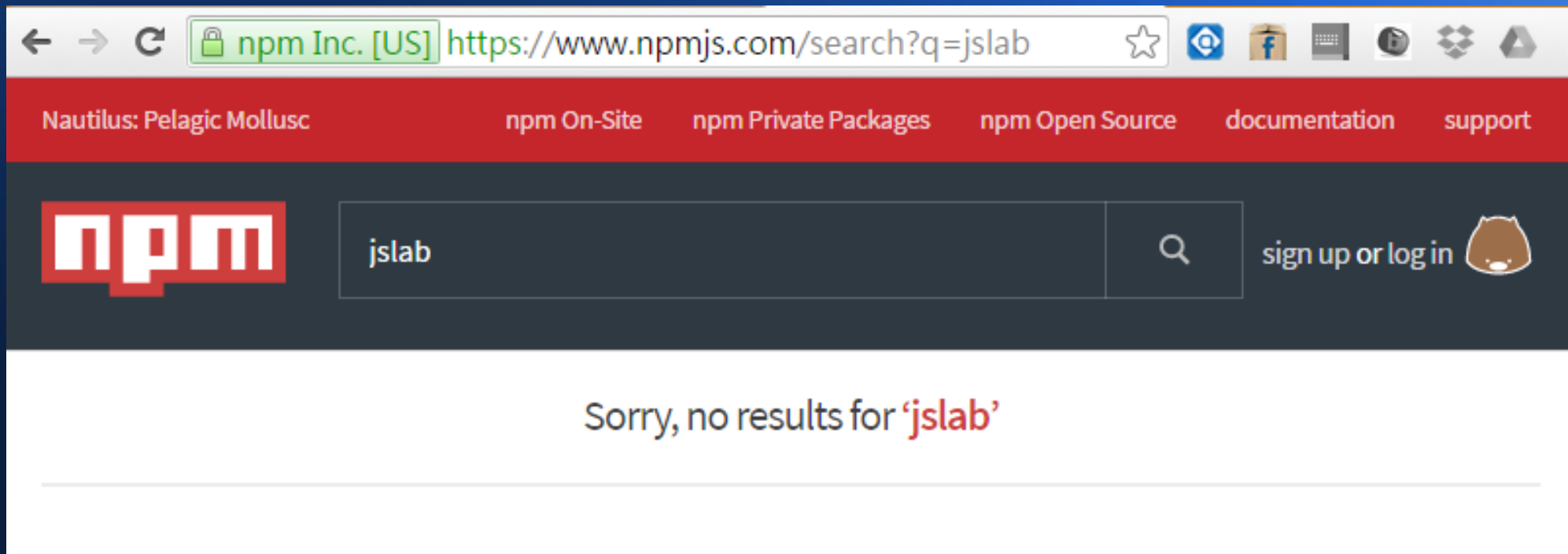
```
1 {
2   "name": "jslab",
3   "version": "0.0.1",
4   "description": "A Scientific Computing Library lib
5   "main": "jslab.js",
6   "dependencies": {
7     "numeric": "^1.2.6"
8   },
9   "devDependencies": {},
10  "scripts": {
11    "test": "jslab_test"
12  },
13  "repository": {
14    "type": "git",
15    "url": "http://ccckmit.github.com/jslab"
16  },
```

這樣、我們就已經準備好

- 可以將 js1ab 套件發佈到 npm 官網上了

但是、請先確定
該套件名稱沒有被別人先申請走

- 否則您的上傳可能會失敗！



接著、我們就可以進入發佈程序

- 將 js1ab 套件發佈到 npm 官網上

但是、在發佈之前

- 你必須先讓 npm 知道你是誰！

所以我要先執行下列指令

```
npm set init.author.name "ccckmit"
```

```
npm set init.author.email "ccckmit@gmail.com"
```

```
npm set init.author.url "http://ccc.nqu.edu.tw"
```

```
npm adduser
```

然後就可以用下列指令發布

- `npm publish ./`



```
C:\n  
D:\jslab>npm publish ./  
+ jslab@0.0.1  
  
D:\jslab>_
```

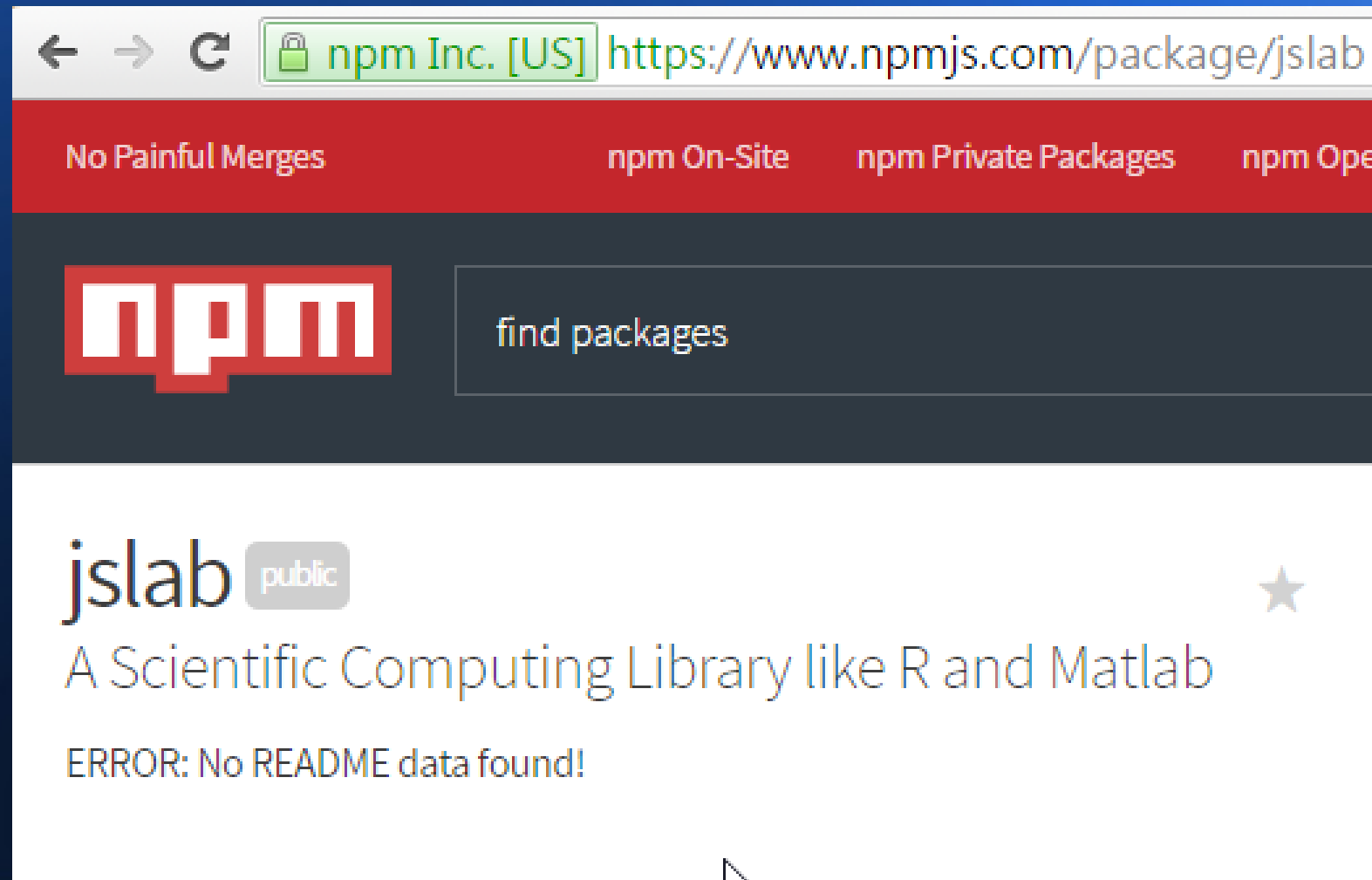
A terminal window with an orange title bar and a black background. The title bar contains a small icon and the text 'C:\'. The terminal shows the command 'D:\jslab>npm publish ./' being executed, followed by the output '+ jslab@0.0.1'. The prompt 'D:\jslab>' is shown again on the next line, followed by an underscore character '_'. The terminal window is positioned in the lower half of the slide.

接著再上 npm 官網查一下

- 看看您的專案是否真的發布成功了



點選 jslab 連結進去檢查看看



您會發現

- 該專案雖然發佈成功，但是卻缺少了 README 的資訊

jslab public



A Scientific Computing Library like R and Matlab

ERROR: No README data found!

沒有 README 資訊

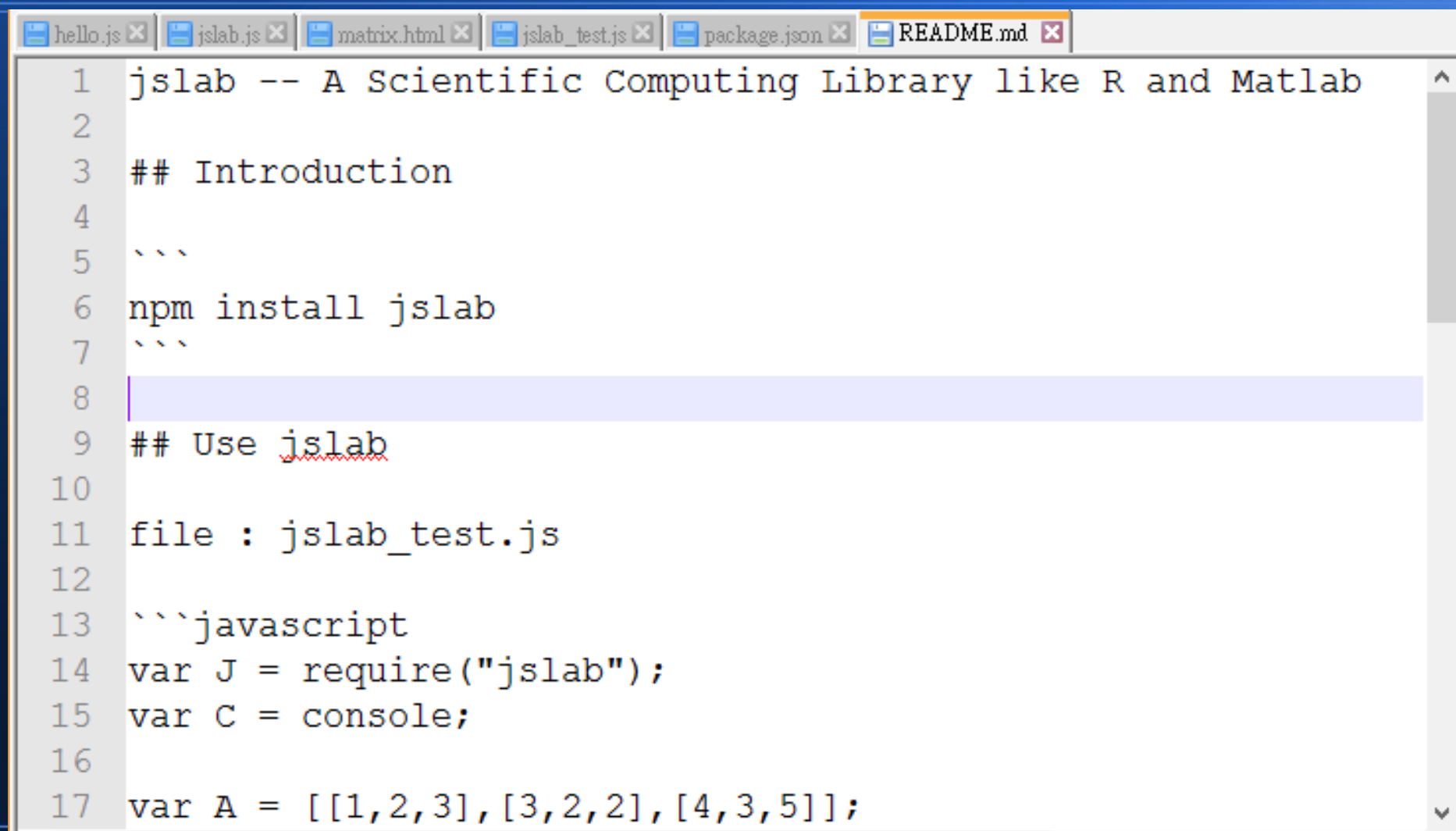
- 別人就不知道怎麼用你的專案

所以我們應該加入 README 資訊

README 是個文字檔

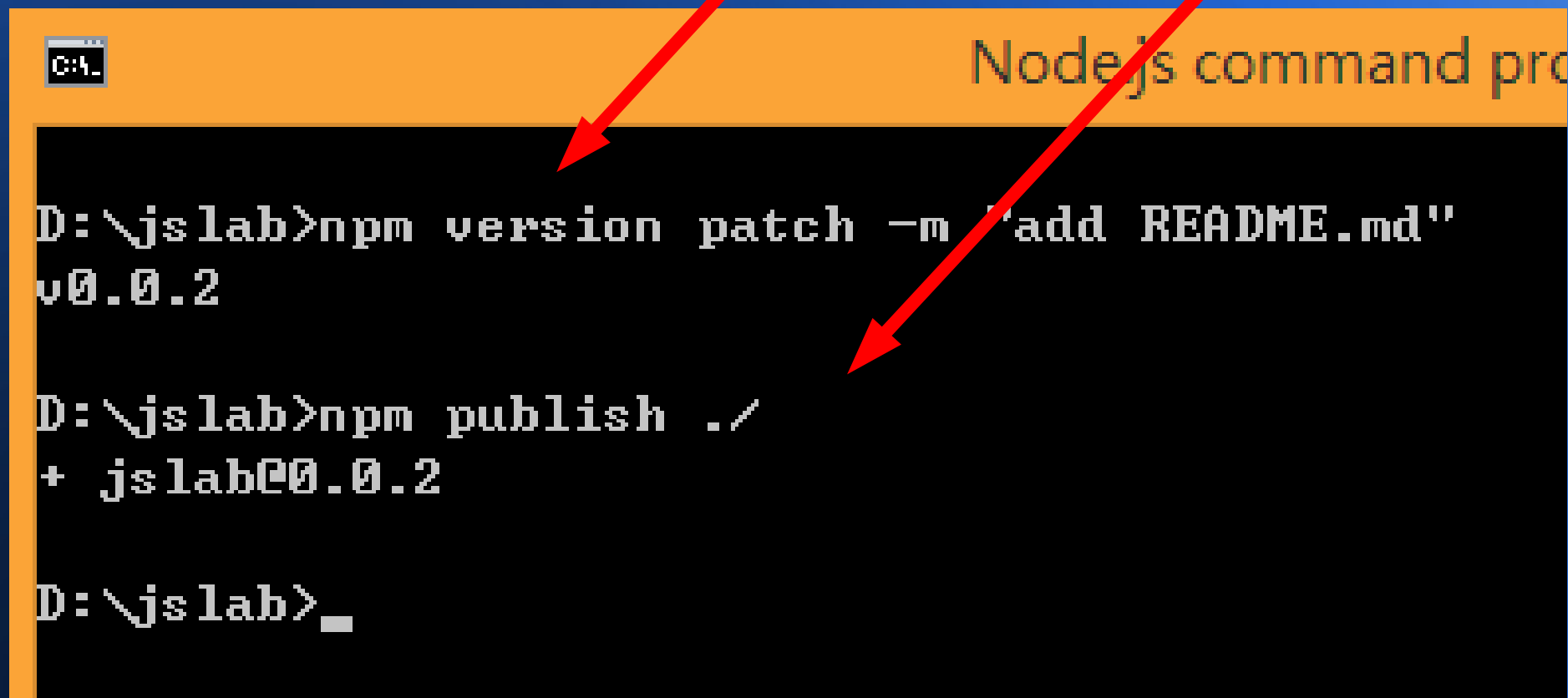
- 現在通常用 markdown 格式寫這種檔案，所以副檔名可以用 .md
- 也就是要寫一個 README.md 檔案

以下是 jslab 的 README.md 檔案



```
1 jslab -- A Scientific Computing Library like R and Matlab
2
3 ## Introduction
4
5 ```
6 npm install jslab
7 ```
8
9 ## Use jslab
10
11 file : jslab_test.js
12
13 ```javascript
14 var J = require("jslab");
15 var C = console;
16
17 var A = [[1,2,3],[3,2,2],[4,3,5]];
```

寫好後就可以更新版本再發佈



A terminal window titled "Nodejs command pro" with a yellow header bar. The terminal shows the following commands and output:

```
D:\jslab>npm version patch -m "add README.md"
v0.0.2

D:\jslab>npm publish ./
+ jslab@0.0.2

D:\jslab>_
```

Two red arrows point from the text "寫好後就可以更新版本再發佈" to the terminal. One arrow points to the `npm version patch` command, and the other points to the `npm publish` command.

再檢查一下就會發現
README 資訊已經放上去



The screenshot shows a web browser window with the address bar displaying 'npm Inc. [US] https://www.npmjs.com/package/jslab'. The page content includes the package name 'jslab' with a 'public' badge and a star icon, followed by the description 'A Scientific Computing Library like R and Matlab'. Below this is an 'Introduction' section with the command 'npm install jslab'. The 'Use jslab' section shows a code example for a file named 'jslab_test.js'.

← → ↻  npm Inc. [US] <https://www.npmjs.com/package/jslab>

jslab public ★

A Scientific Computing Library like R and Matlab

Introduction

```
npm install jslab
```

Use jslab

file: jslab_test.js

```
var J = require("jslab");
var C = console;

var A = [[1,2,3],[3,2,2],[4,3,5]];
C.log("A=\n", J.str(A));

var iA = J.inv(A);
```

然後、當別人想要用 這個 **jslab** 套件時

- 就可以使用下列指令
 - `npm install jslab`
- 輕輕鬆鬆的安裝你的套件來用了。

現在、您應該已經學會

- 如何使用別人的套件，以及
如何將套件發佈在 npm 官網上！

當然

- 這份投影片只是個起點
- 並不是全部！

或許您還可以進一步學習

如何發佈《網站型 npm 專案》，像是

- 設定 package.json 中的 bin 欄位
- 將套件用 browserify 或 webpack 轉為網站用的 .js 程式
- 正確放置《網頁資料夾》與撰寫《伺服器程式》
- 如何使用 express, koa 套件撰寫《伺服器程式》
- 如何存取 mongodb, postgresSQL 資料庫
- 如何架站或將網站發佈在 Heroku, MS. Azure, Amazon EC2 等雲端網站上等等。

關於這些主題

- 就請各位讀者再去參考相關資源了！

以上

- 就是我們今天的十分鐘系列！

希望

- 這次的十分鐘系列
能讓您真正學會
撰寫與發佈 npm 模組的方法！

我們下次見囉！

Bye Bye !