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

- 我就更不瞭解它了！

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 …

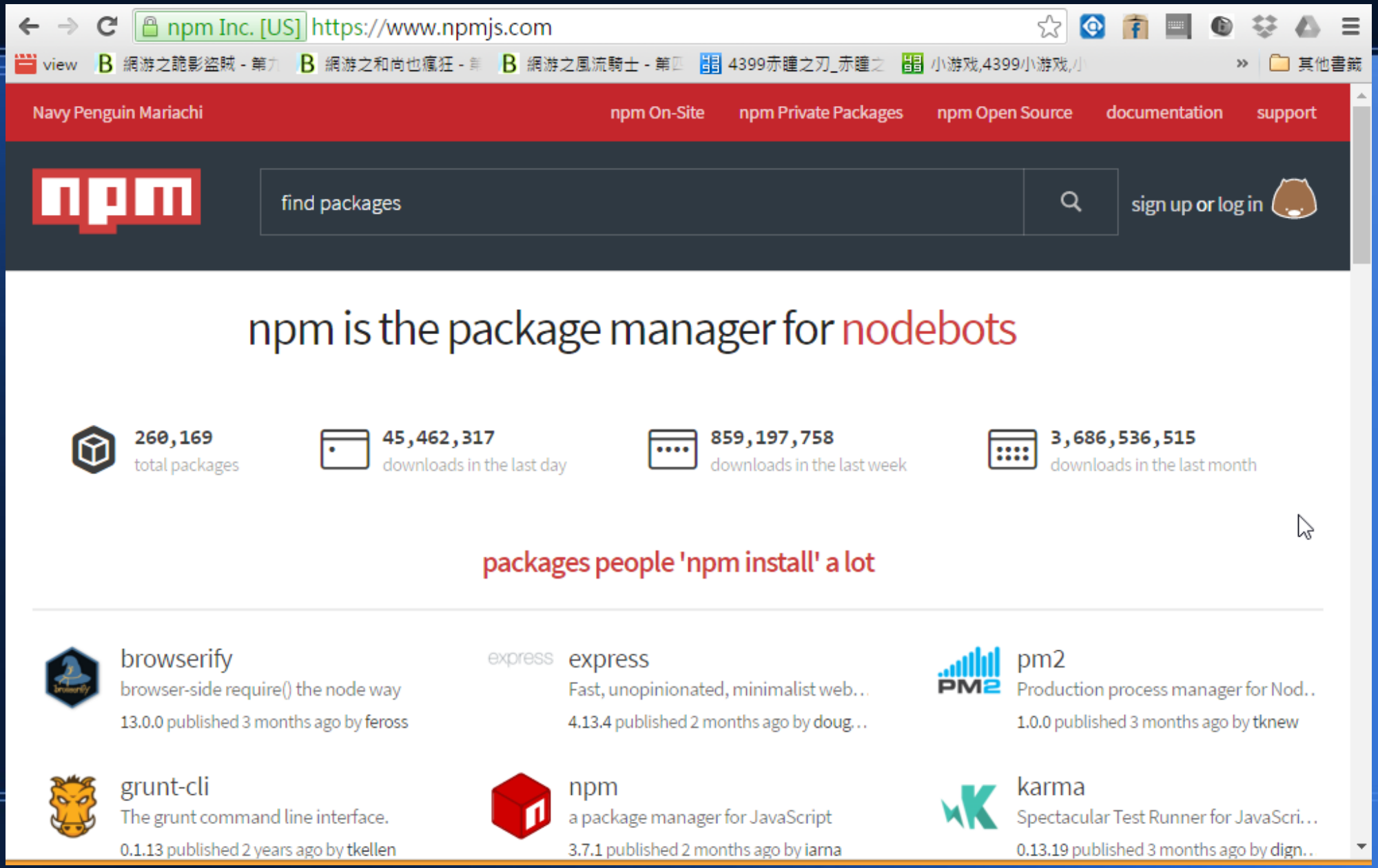
即使只把焦點鎖定在 node. js 領域

- 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 ?


npm Inc. [US] https://www.npmjs.com
Navy Penguin Mariachi
npm On-Site
npm Private Packages
npm Open Source
documentation
support
npm
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！

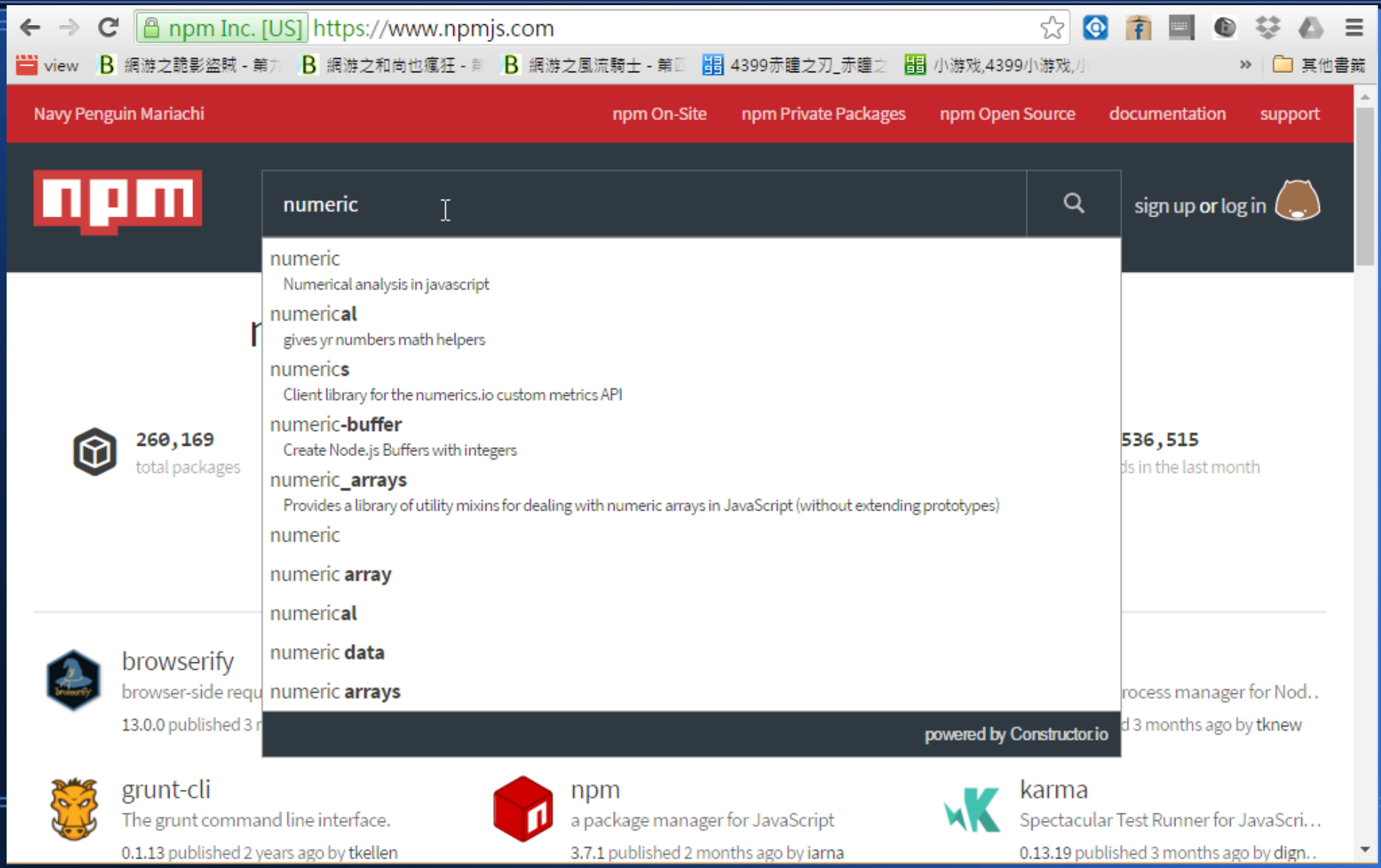
想使用 npm 套件

- 首先您得上 npm 網站去查詢
- 你想要用的套件！

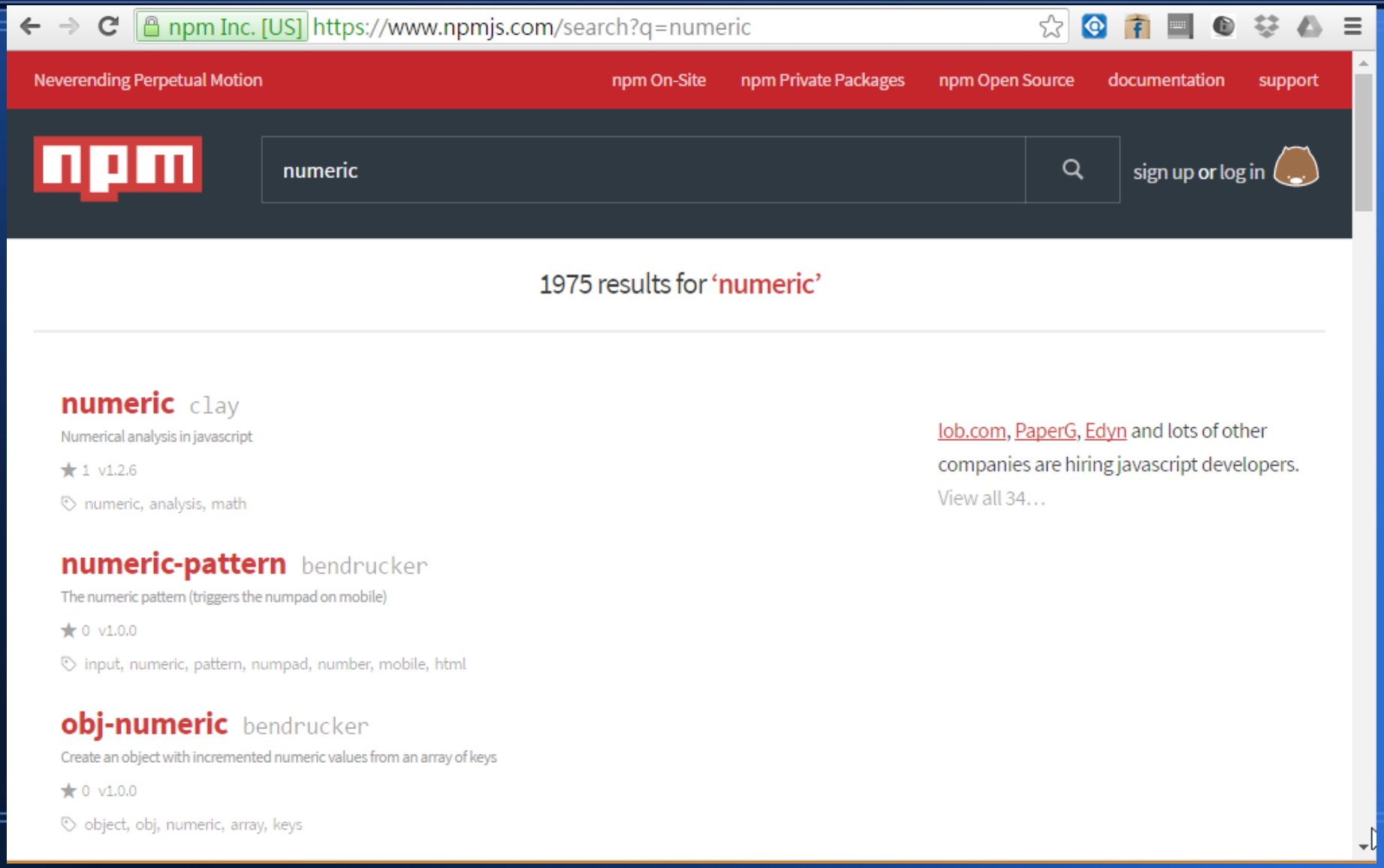
舉例而言

- 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！

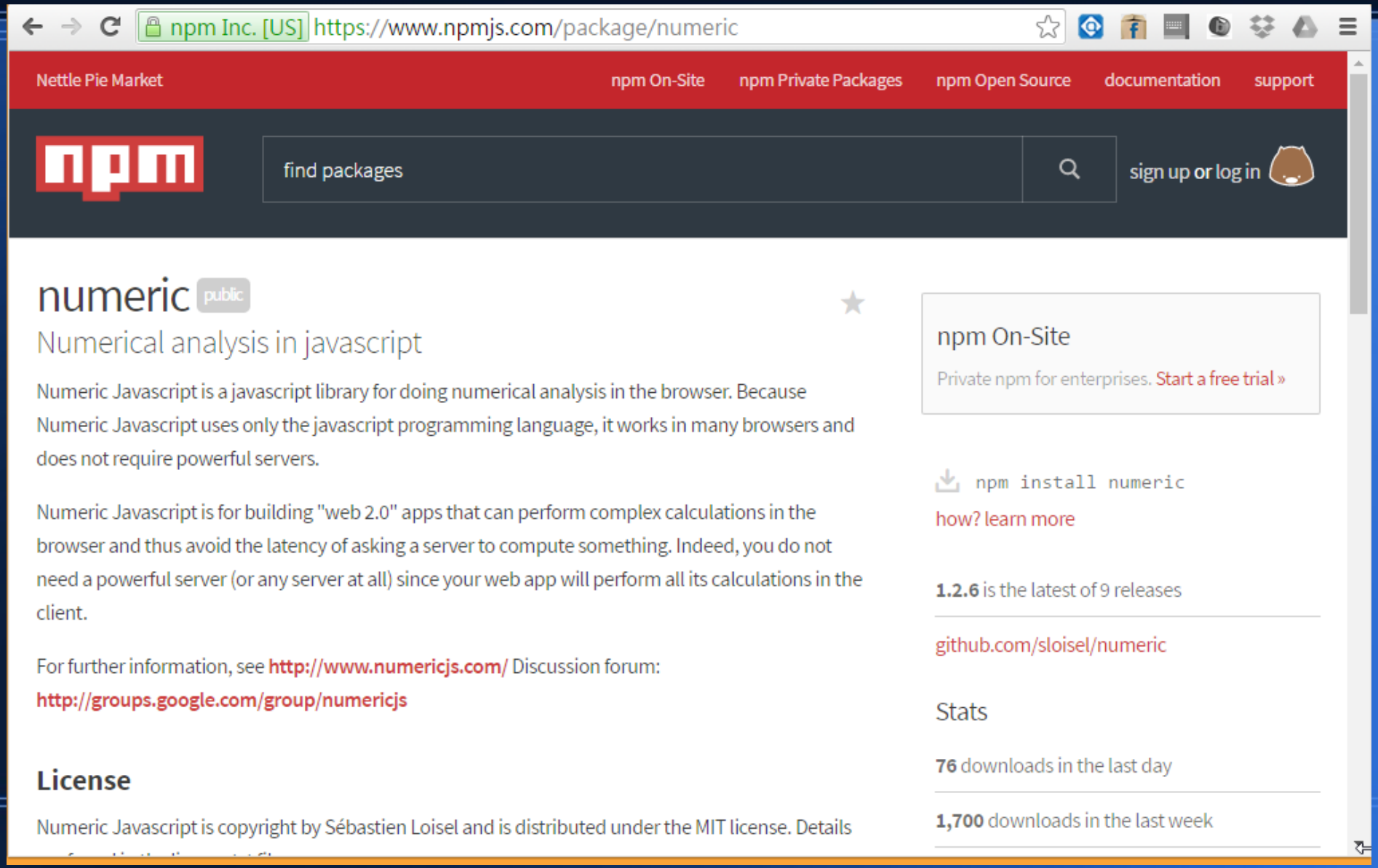
您可以搜尋該套件


npm Inc. [US] https://www.npmjs.com
Navy Penguin Mariachi
npm On-Site
npm Private Packages
npm Open Source
documentation
support
numeric
sign up or log in
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
numeric array
numerical
numeric data
numeric arrays
powered by Constructor.io
260,169
total packages
536,515
browserify
13.0.0 published 3
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..

找到之後


npm Inc. [US] https://www.npmjs.com/search?q=numeric
Neverending Perpetual Motion
npm On-Site
npm Private Packages
npm Open Source
documentation
support
npm
numeric
sign up or log in
1975 results for 'numeric'
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
lob.com, PaperG, Edyn and lots of other companies are hiring javascript developers.
View all 34…

看看套件的描述


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

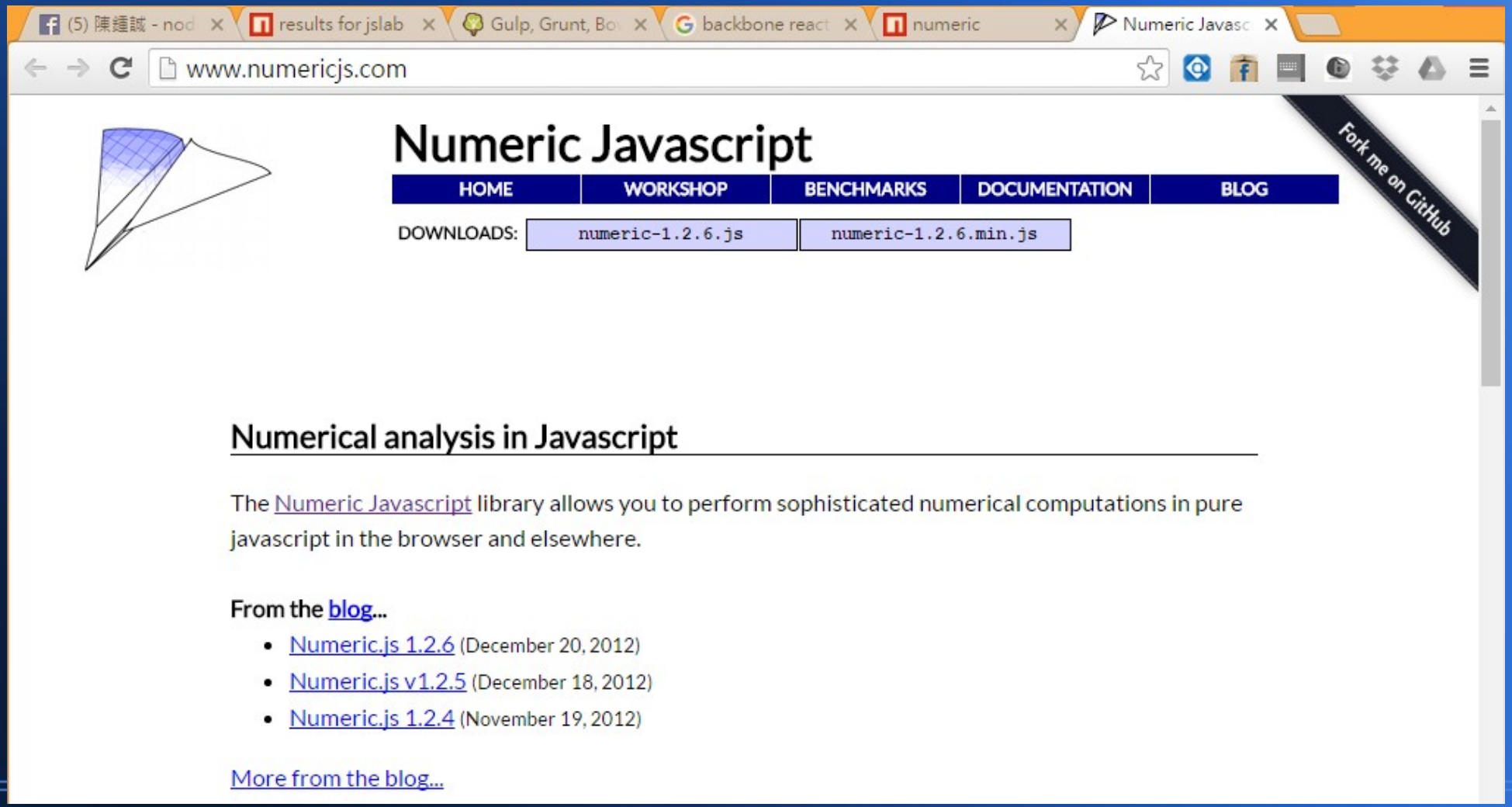
有些比較好的套件

- 還會有官方網站！

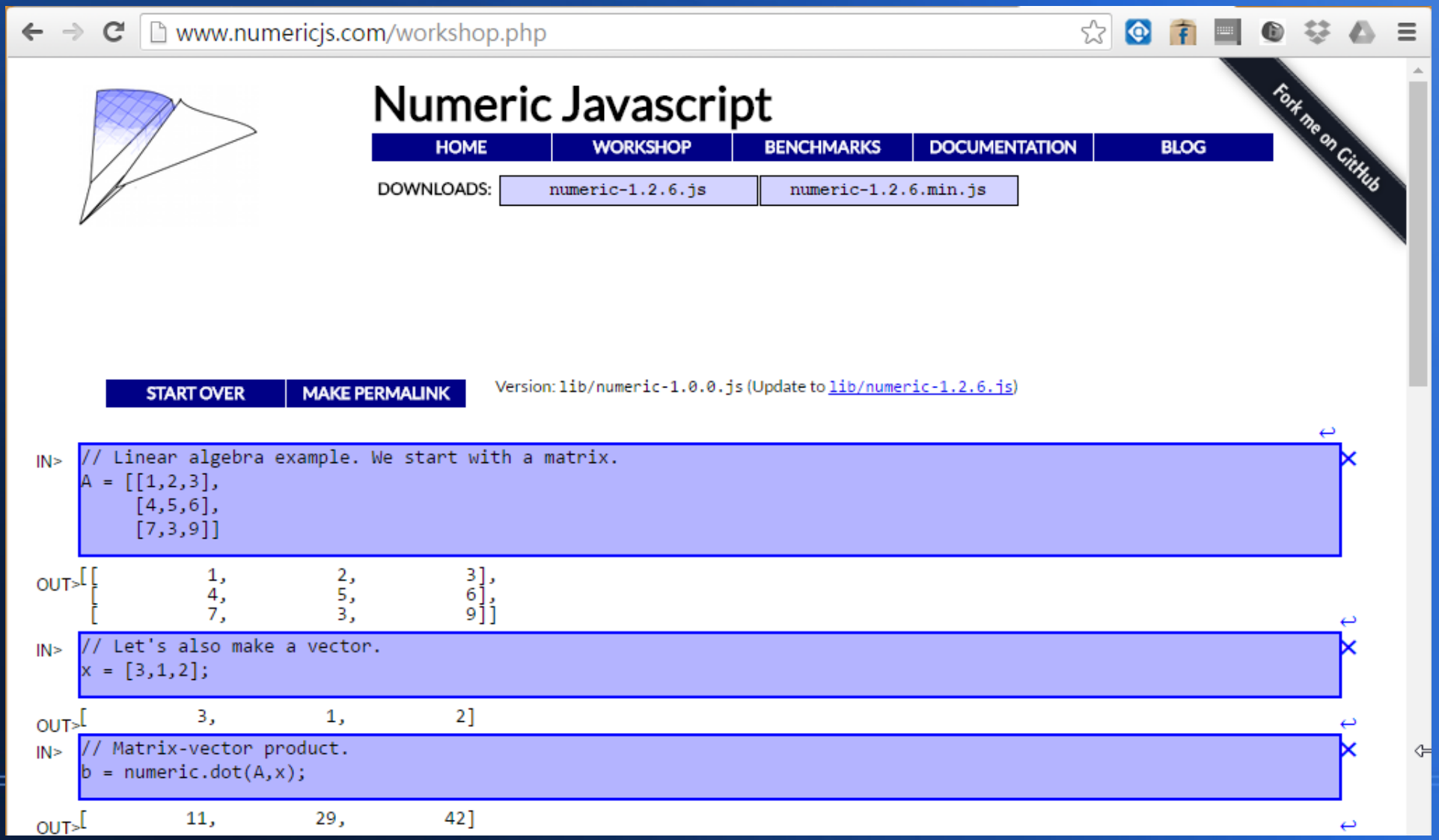
For further information, see **http://www.numericjs.com/** Discussion forum:
http://groups.google.com/group/numericjs

於是你可以去看看

- 該套件的詳細內容！

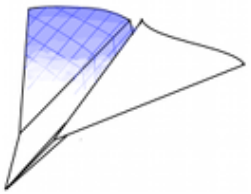


以及使用說明


www.numericjs.com/workshop.php
Numeric Javascript
HOME
WORKSHOP
BENCHMARKS
DOCUMENTATION
BLOG
Fork me on GitHub
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]

還有函式庫定義等等

www.numericjs.com/documentation.html



Numeric Javascript

Fork me on GitHub

HOME | WORKSHOP | BENCHMARKS | DOCUMENTATION | BLOG

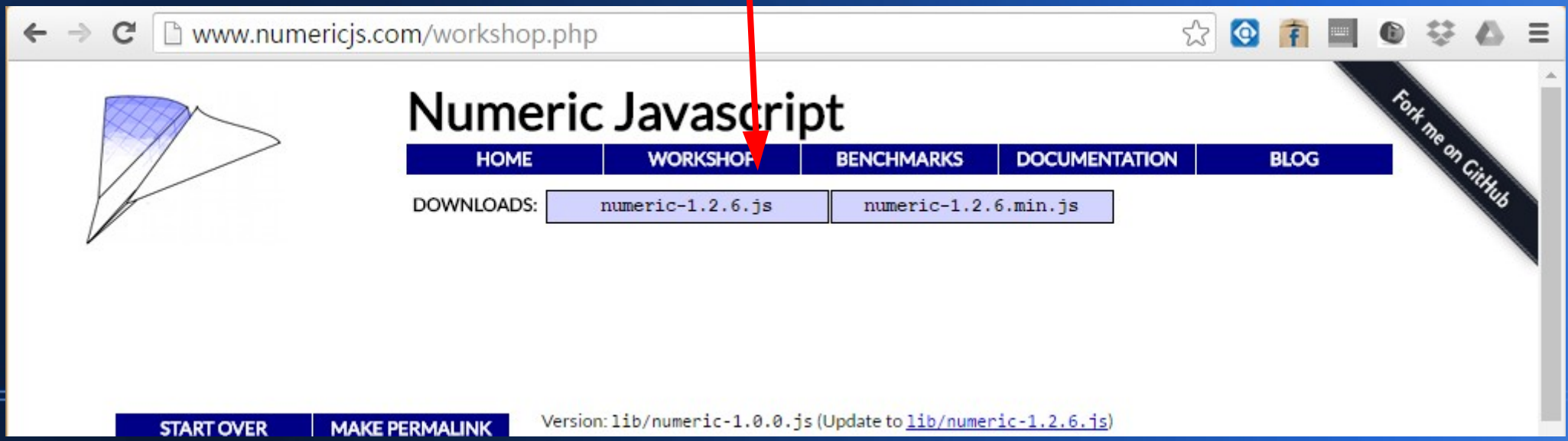
DOWNLOADS: numeric-1.2.6.js | numeric-1.2.6.min.js

Reference card for the `numeric` module

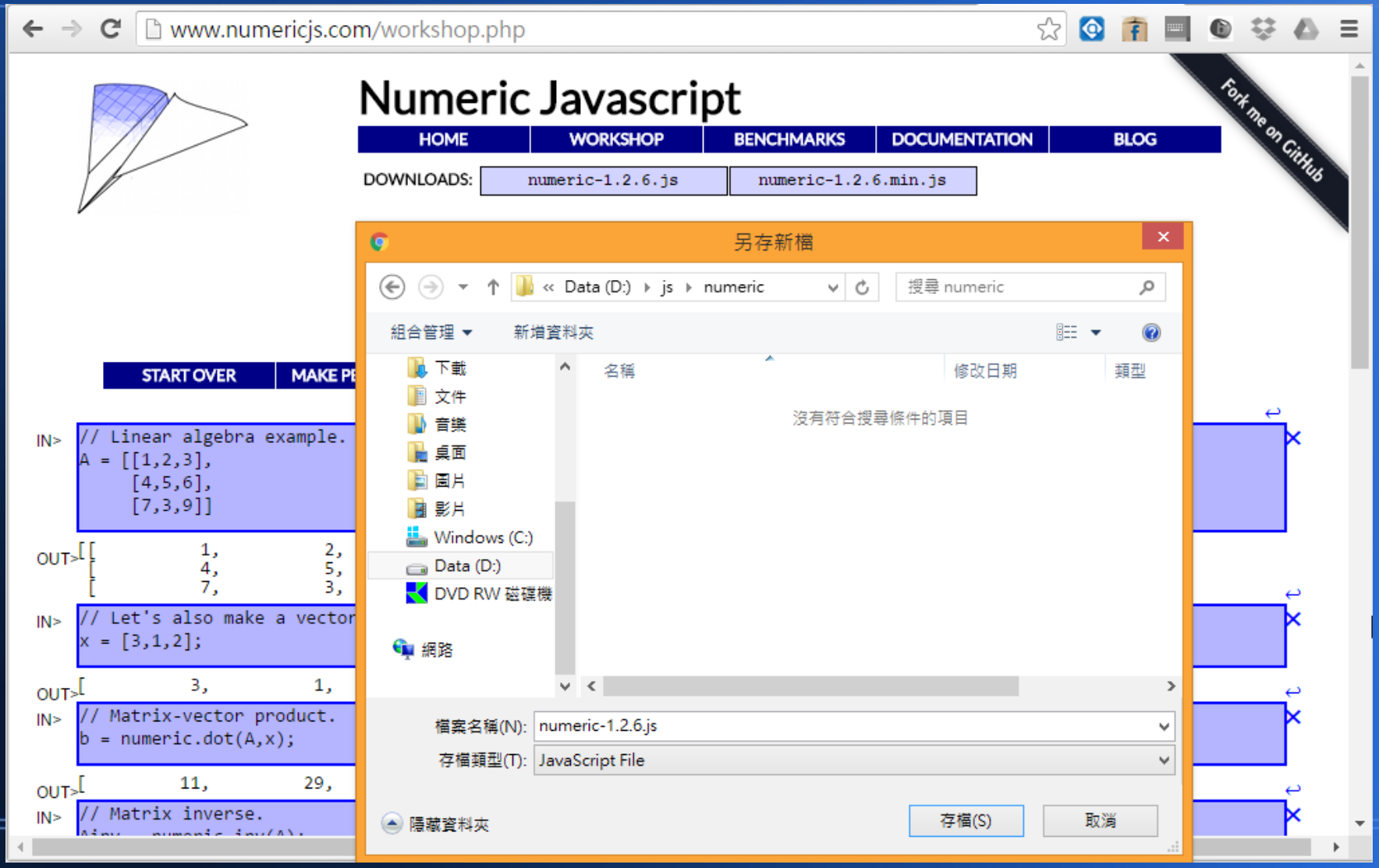
Function	Description	Function	Description	Function	Description
abs	Absolute value	dot	Matrix-Matrix, Matrix-Vector and Vector-Matrix product	round	Pointwise Math.round(x)
acos	Arc-cosine			rrshift	Pointwise x>>>y
add	Pointwise sum x+y	eig	Eigenvalues and eigenvectors	rrshifteq	Pointwise x>>>=y
addeq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

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！

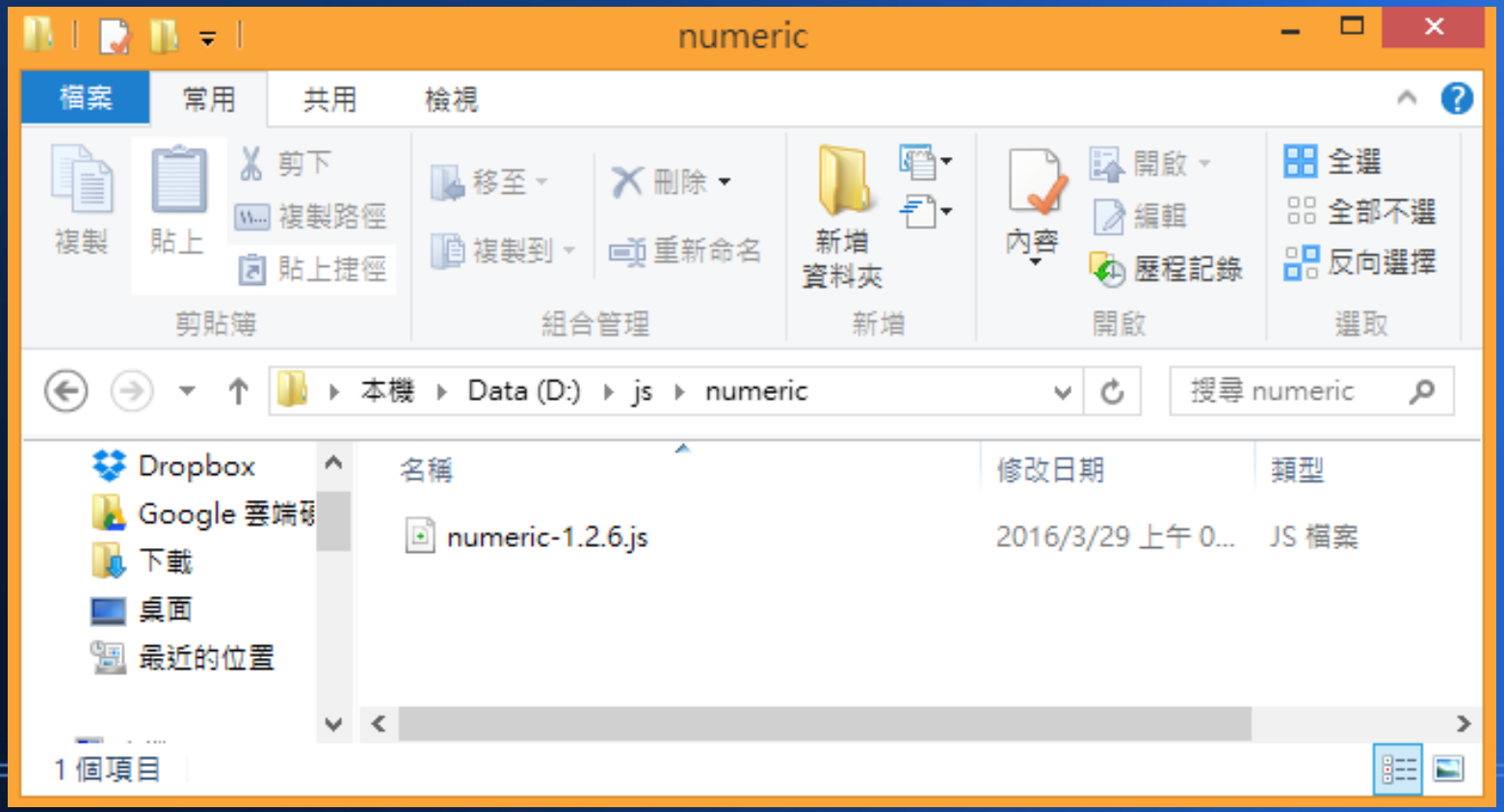


您可以按右鍵選擇下載後存檔使用



但問題是

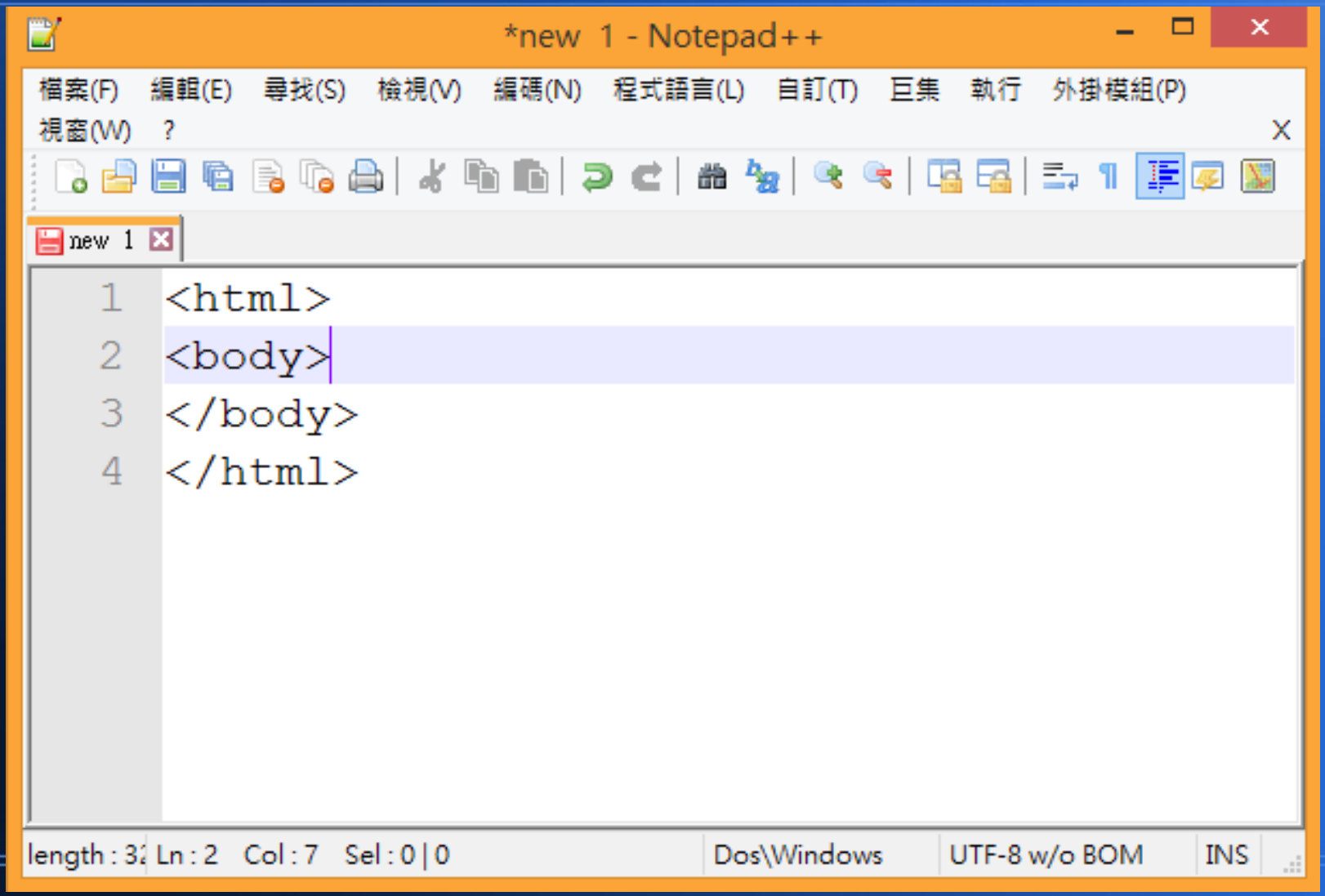
- 下載完 . js 檔案後要怎麼用呢？



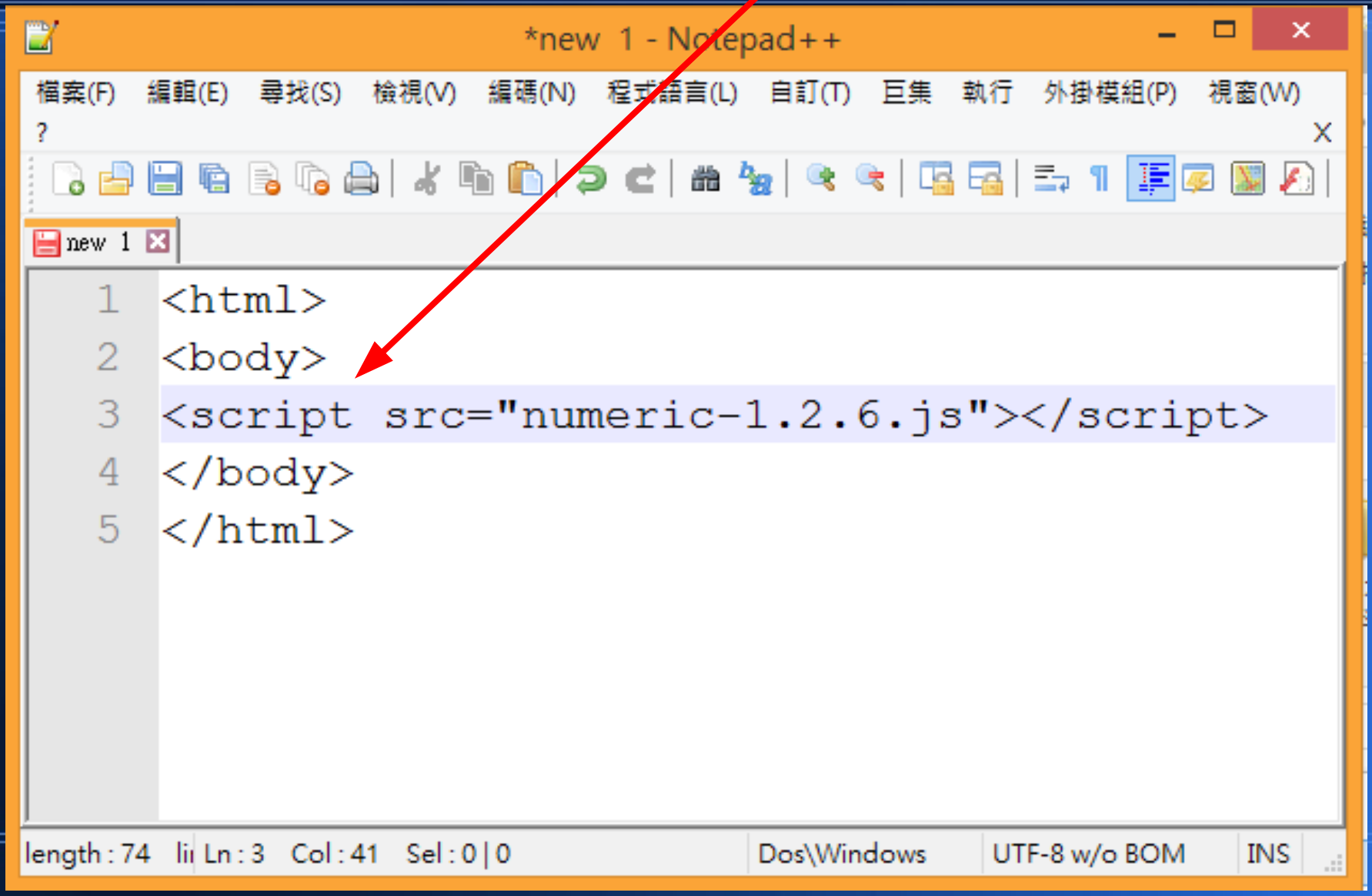
對於這種 .js 檔案

- 通常是放在網頁內使用的！

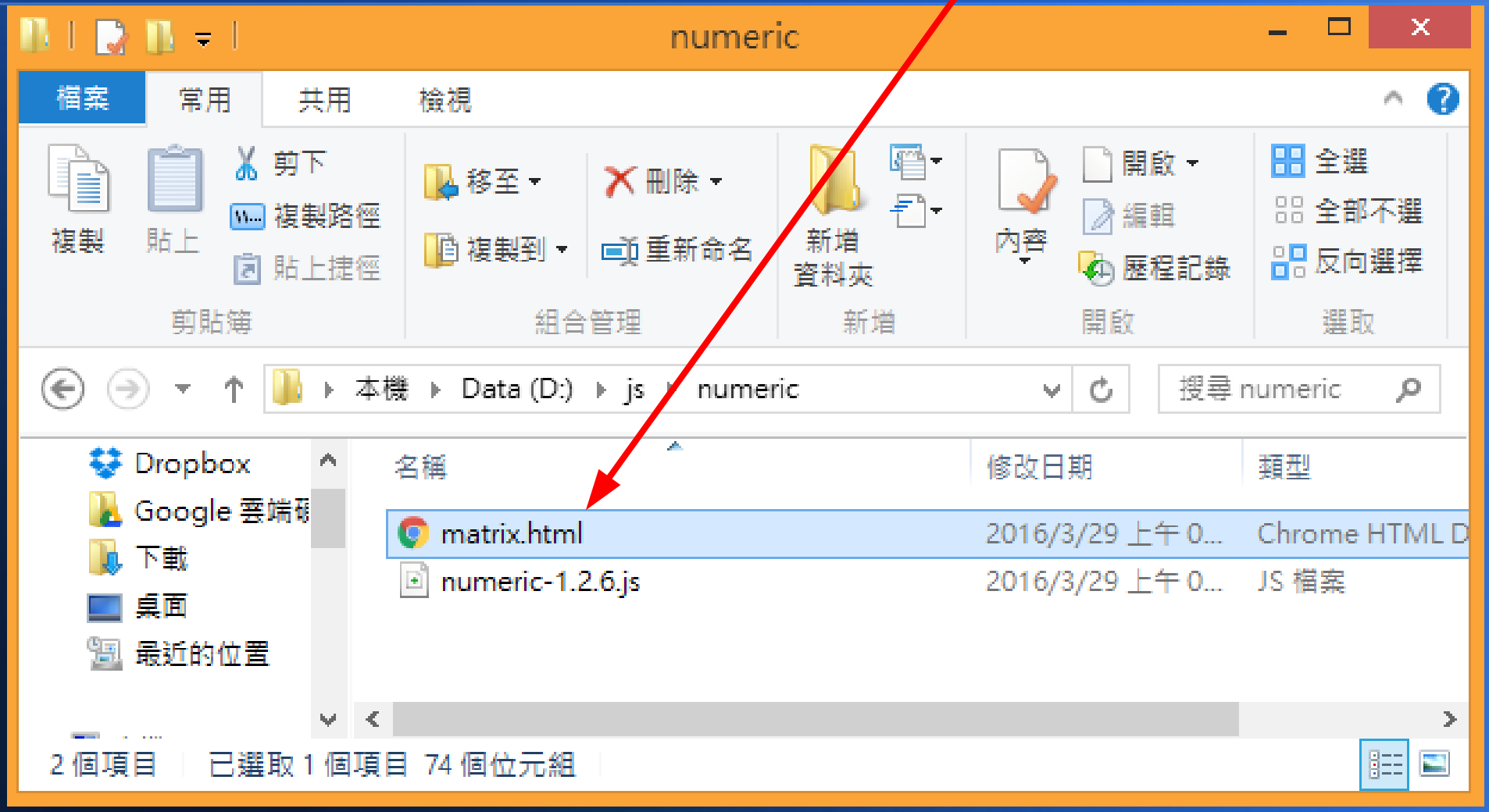
您必須要先寫個 html 網頁



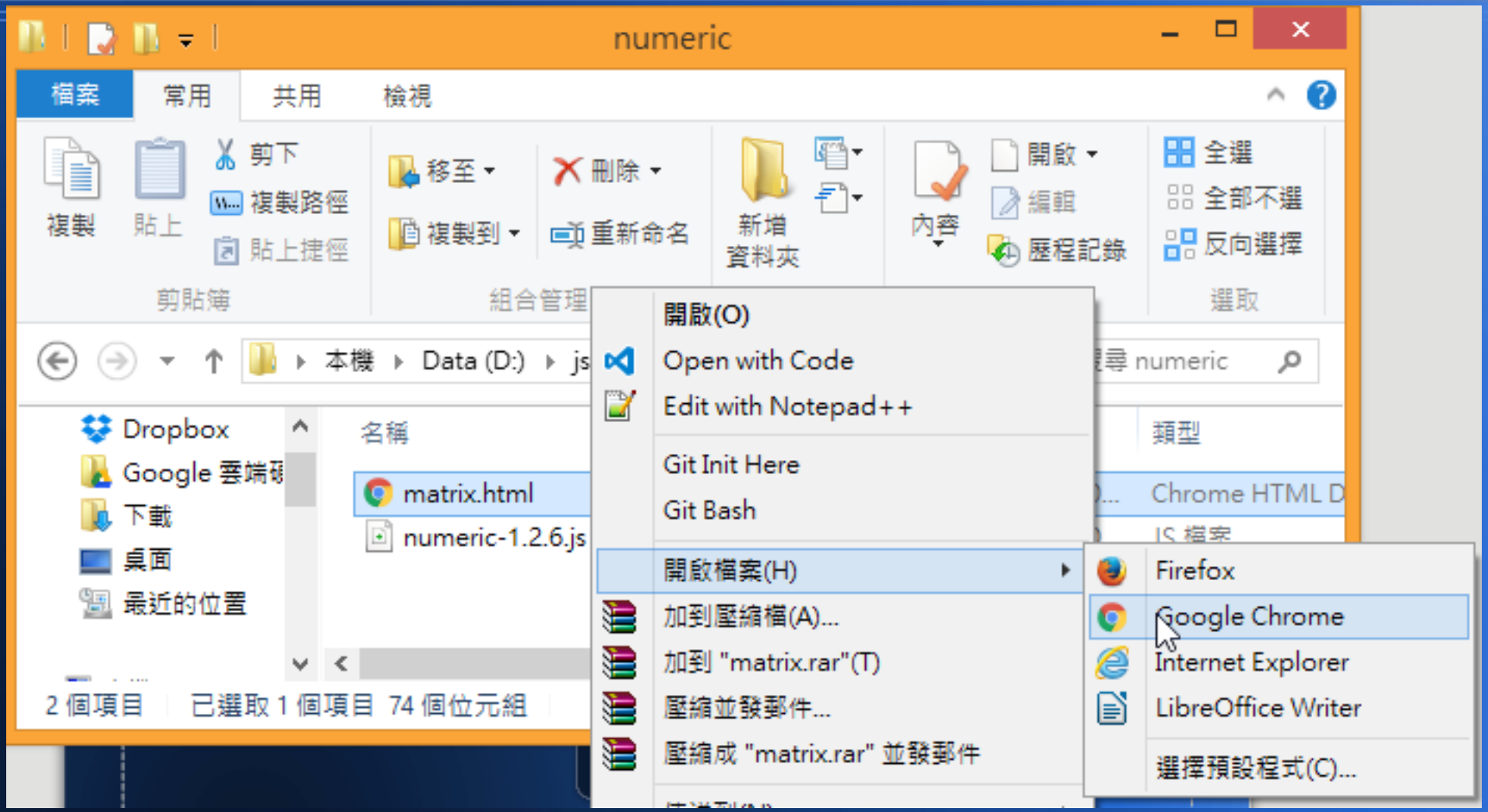
記得加上 script 引用



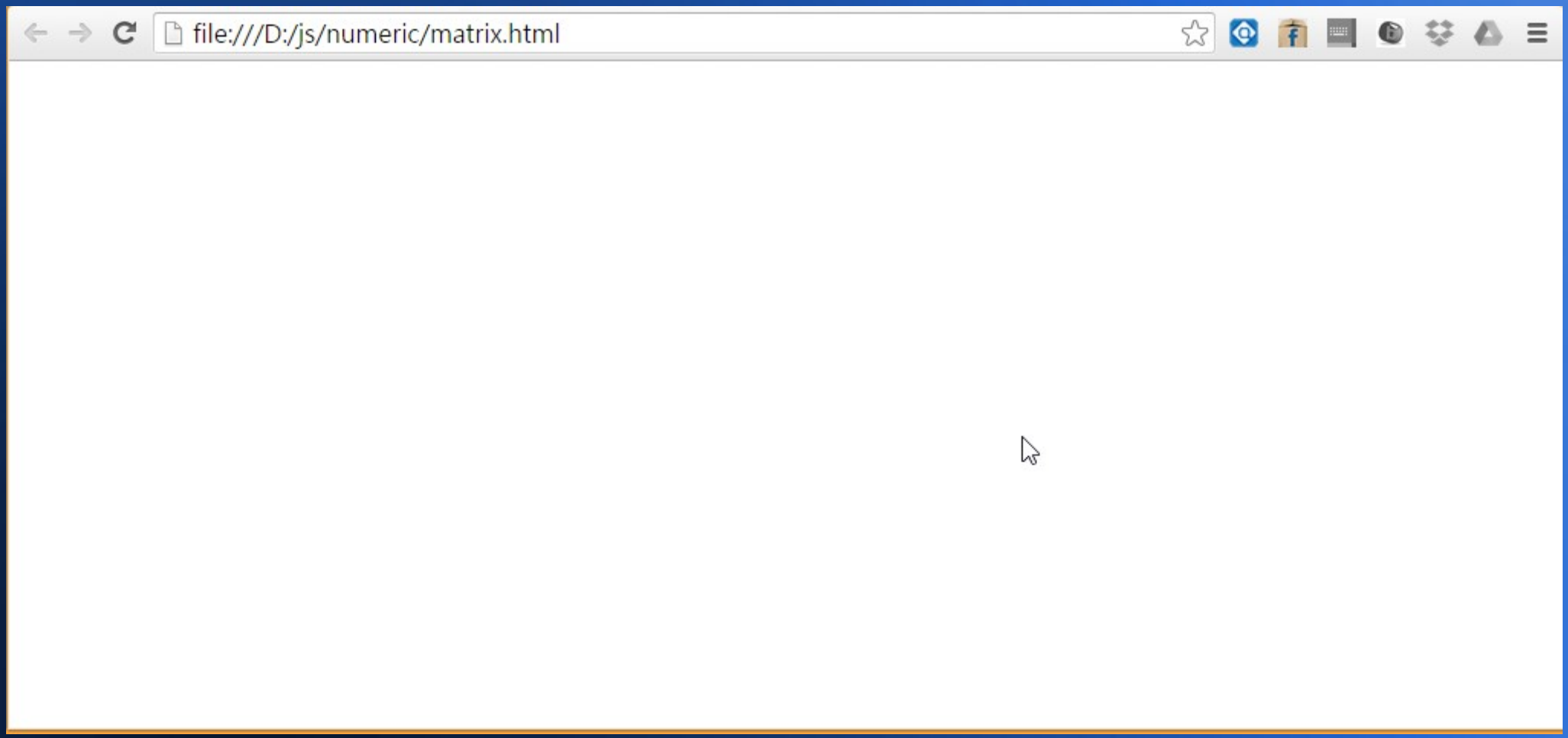
然後就可以存檔


numeric
檔案
常用
共用
檢視
複製
貼上
剪下
複製路徑
貼上捷徑
剪貼簿
移至
複製到
刪除
重新命名
組合管理
新增資料夾
新增
內容
開啟
編輯
歷程記錄
開啟
全選
全部不選
反向選擇
選取
本機 ▸ Data (D:) ▸ js ▸ numeric
搜尋 numeric
Dropbox
Google 雲端硬
下載
桌面
最近的位置
名稱
修改日期
類型
matrix.html
2016/3/29 上午 0...
Chrome HTML D
numeric-1.2.6.js
2016/3/29 上午 0...
JS 檔案
2 個項目
已選取 1 個項目 74 個位元組

接著用 Chrome 瀏覽器打開 html 檔



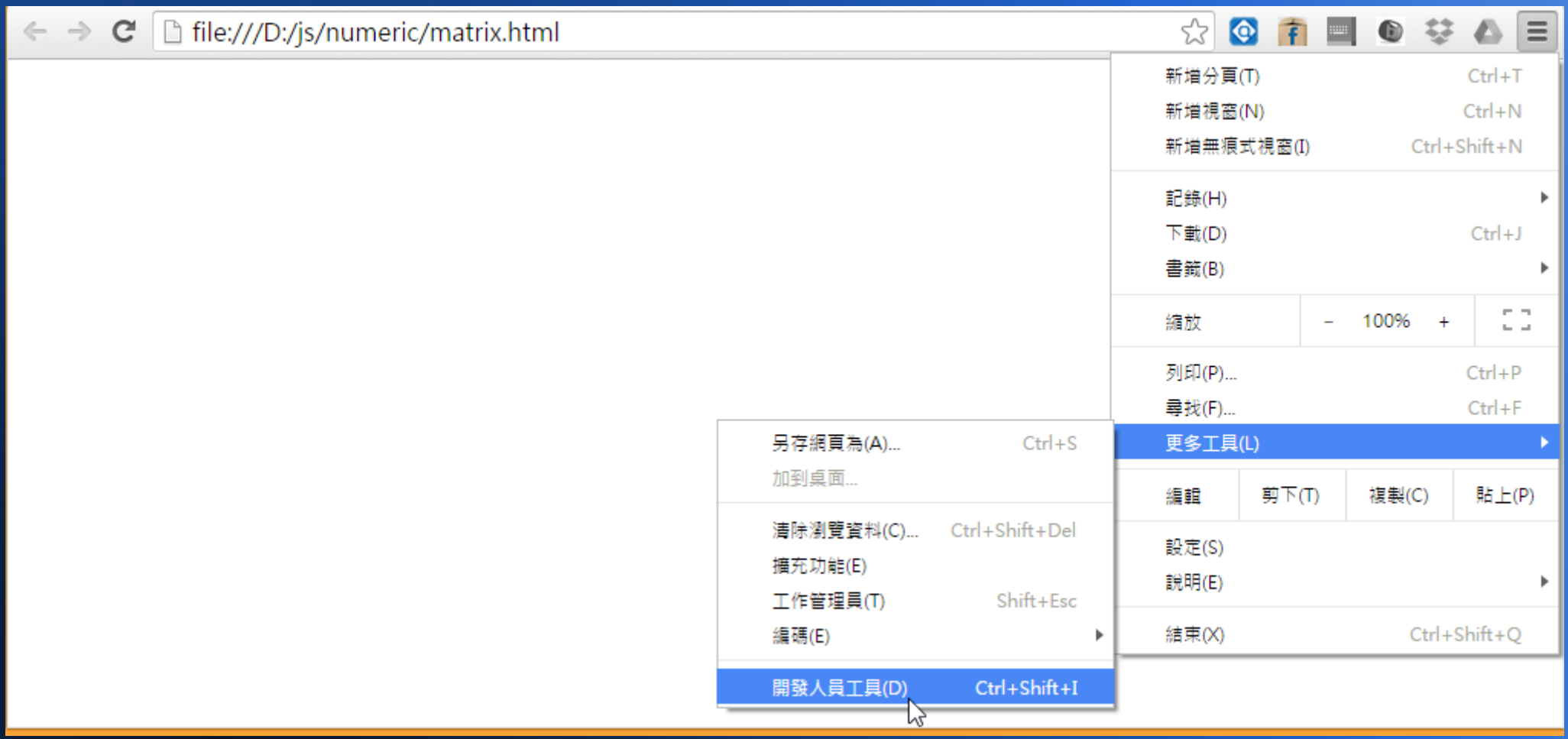
您會看到空空如也！



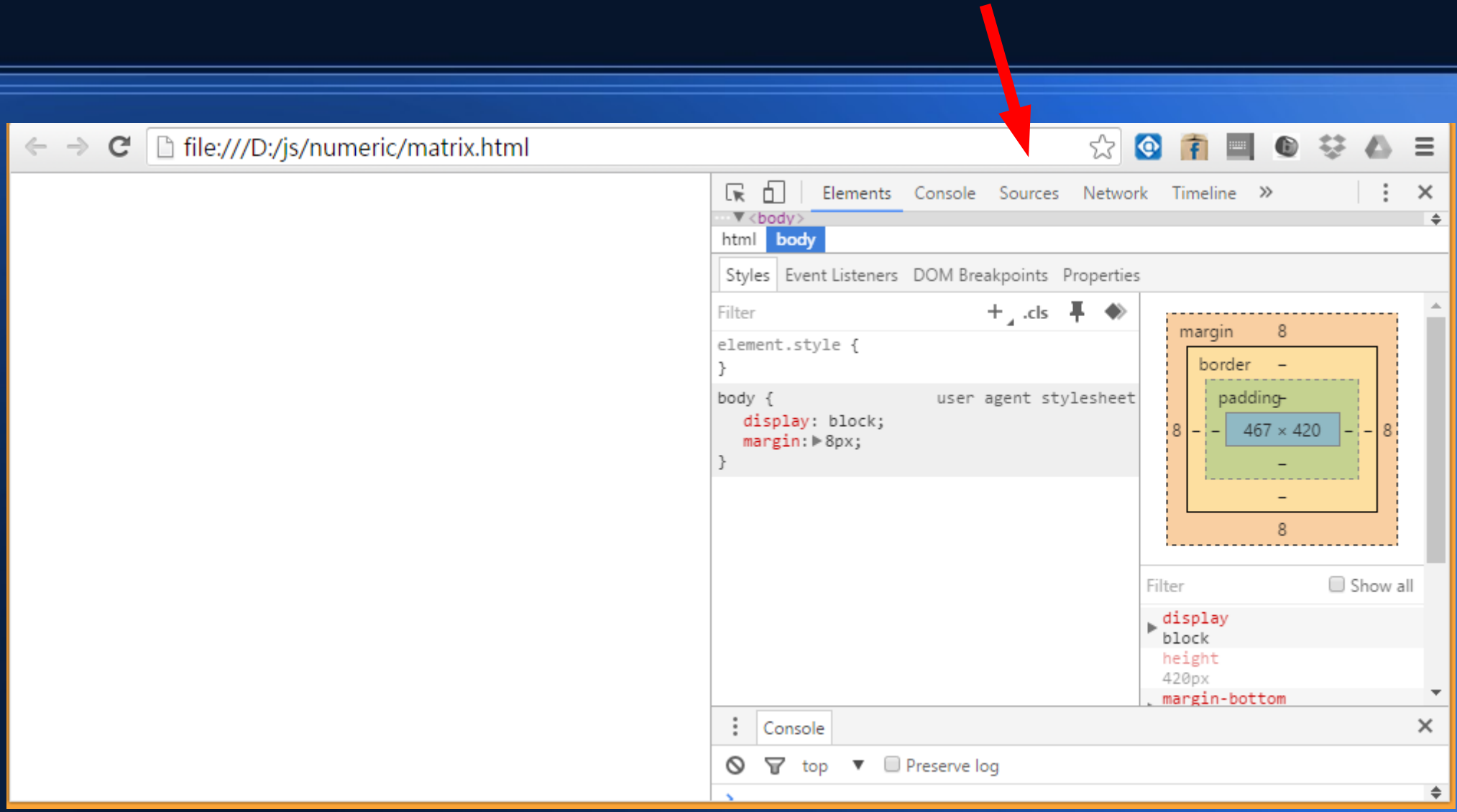
但是沒關係

- 我們有密技！

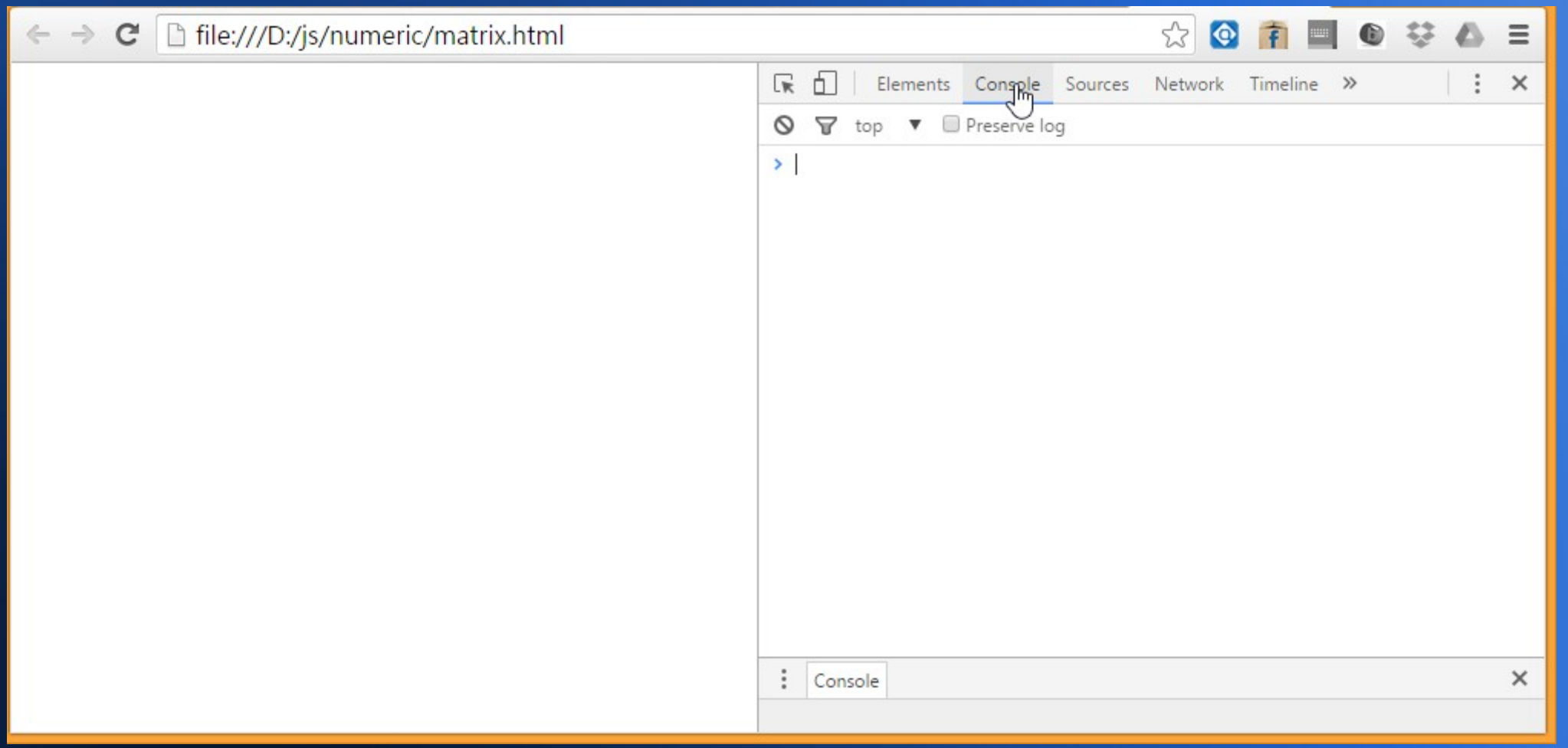
請選《更多工具 / 開發人員工具》



這時您會看到有個窗框跑出來

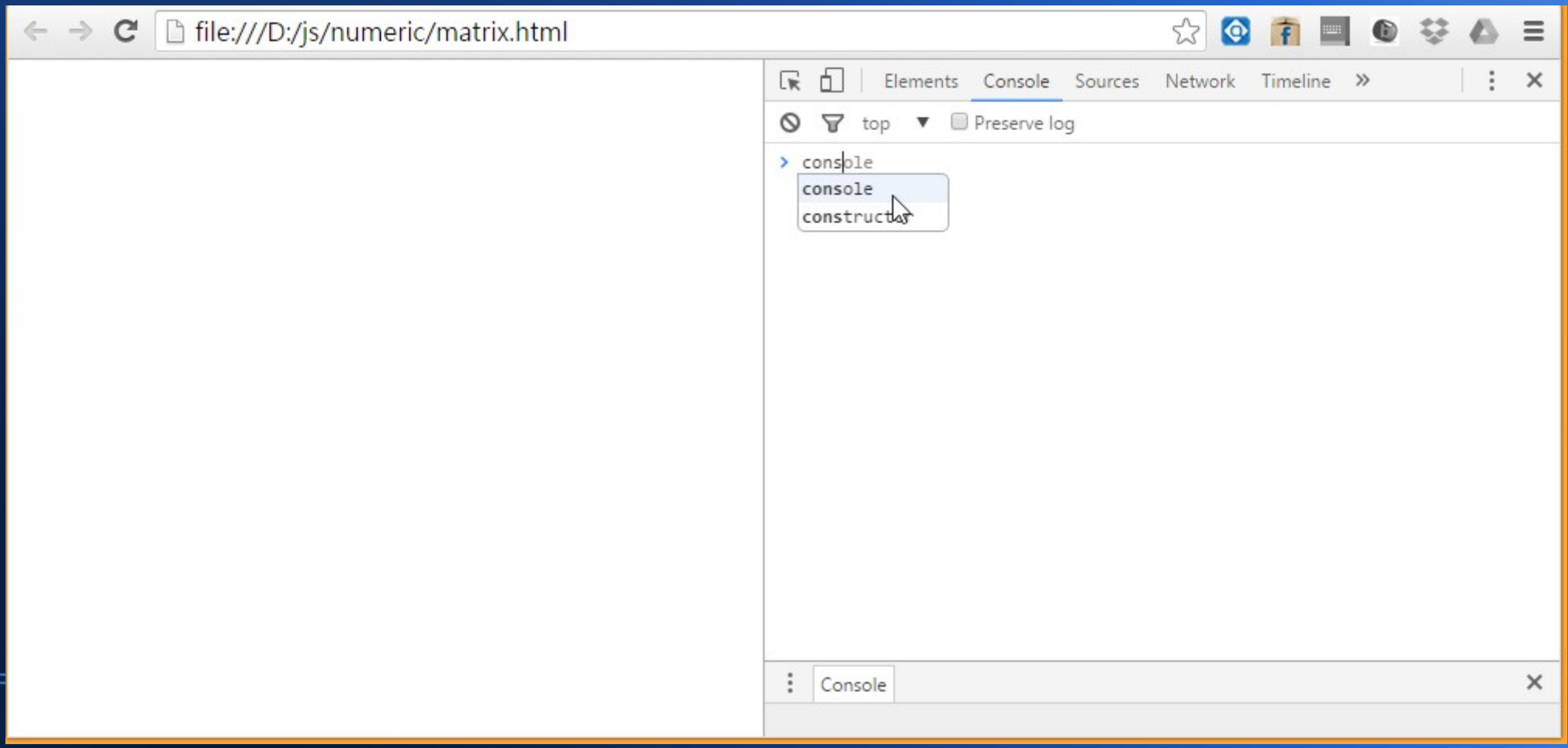


此時您可以選取 console

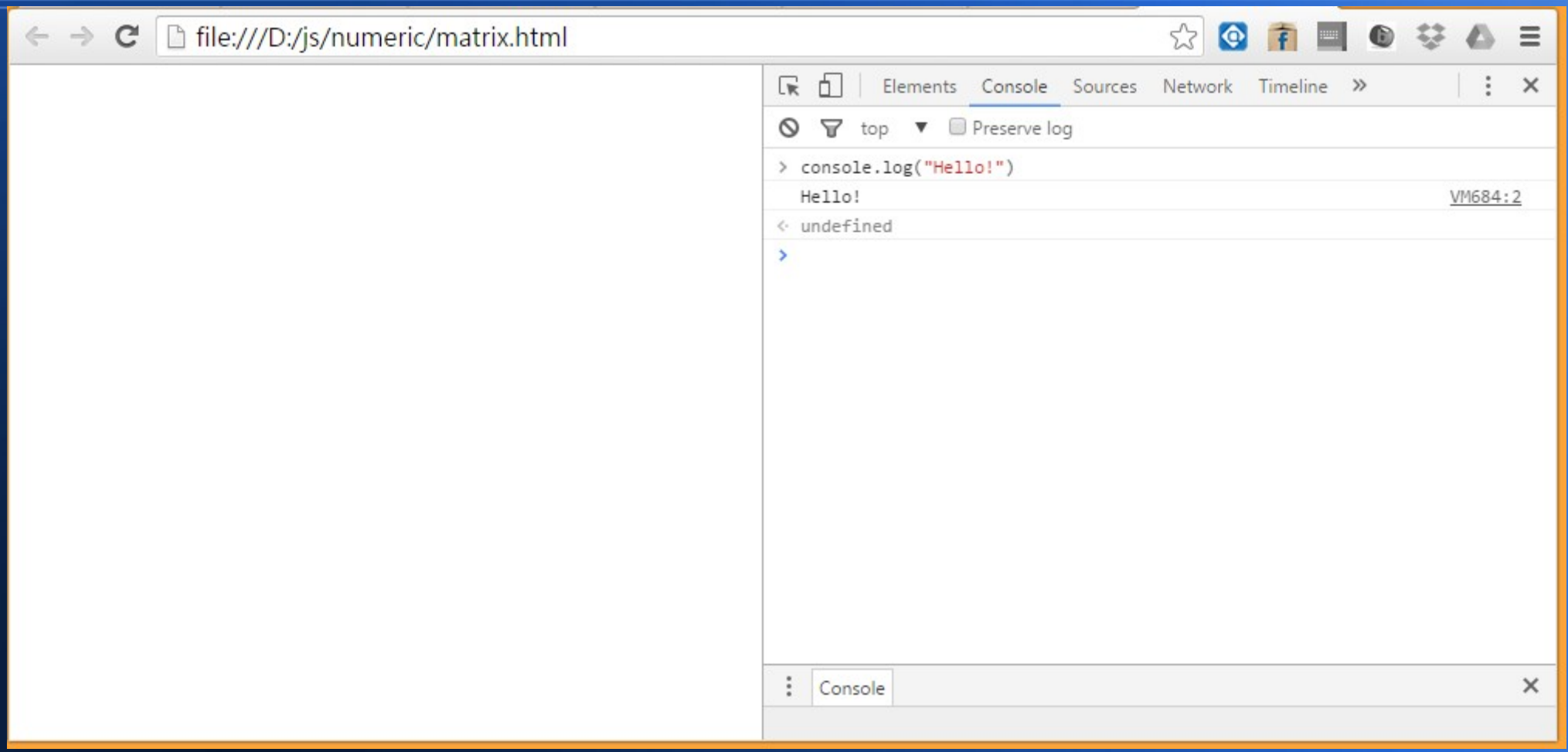


該 console 可以輸入指令

- 而且有很好的提示！

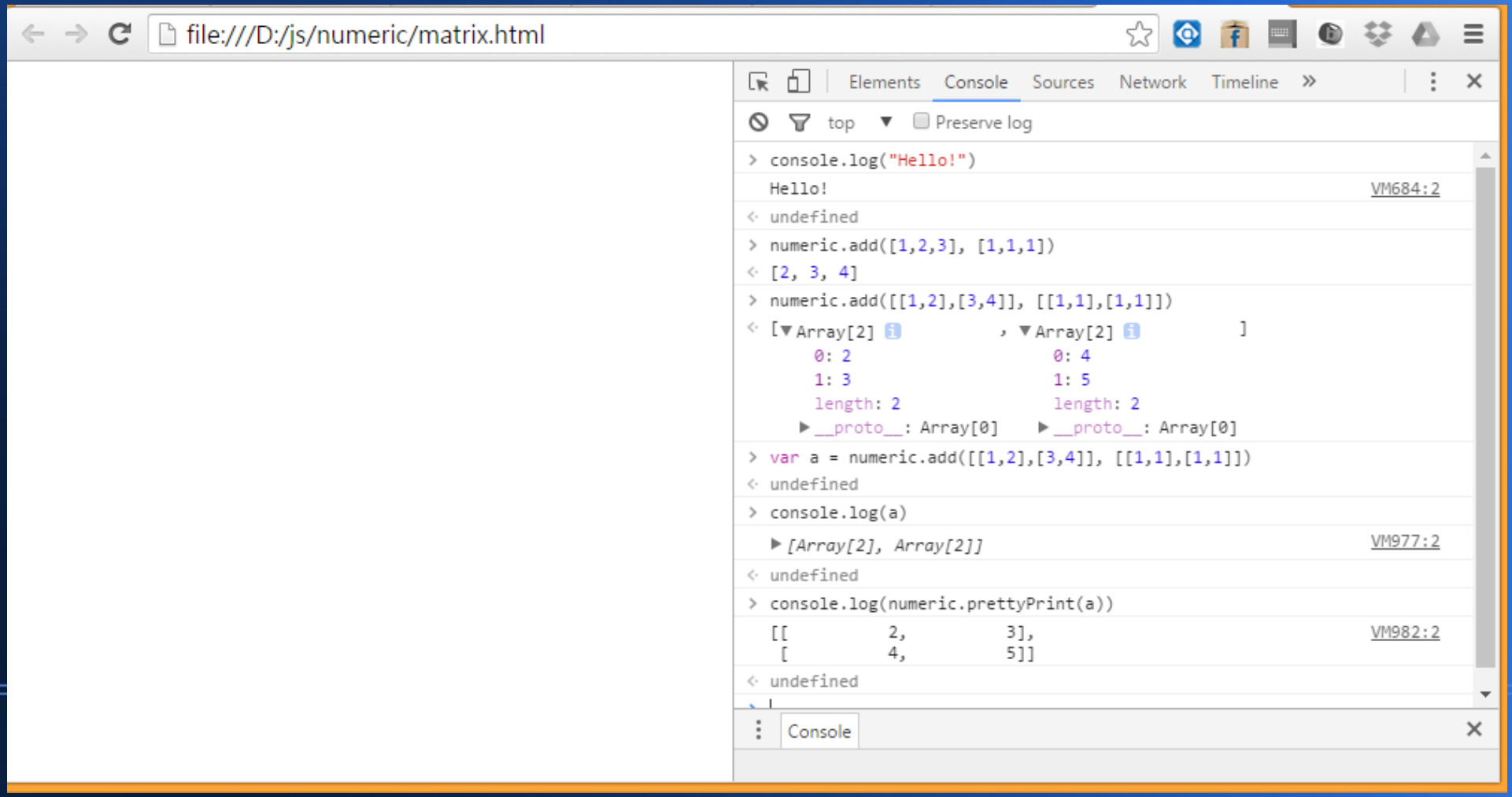


您可以把它當交談式程式環境來用



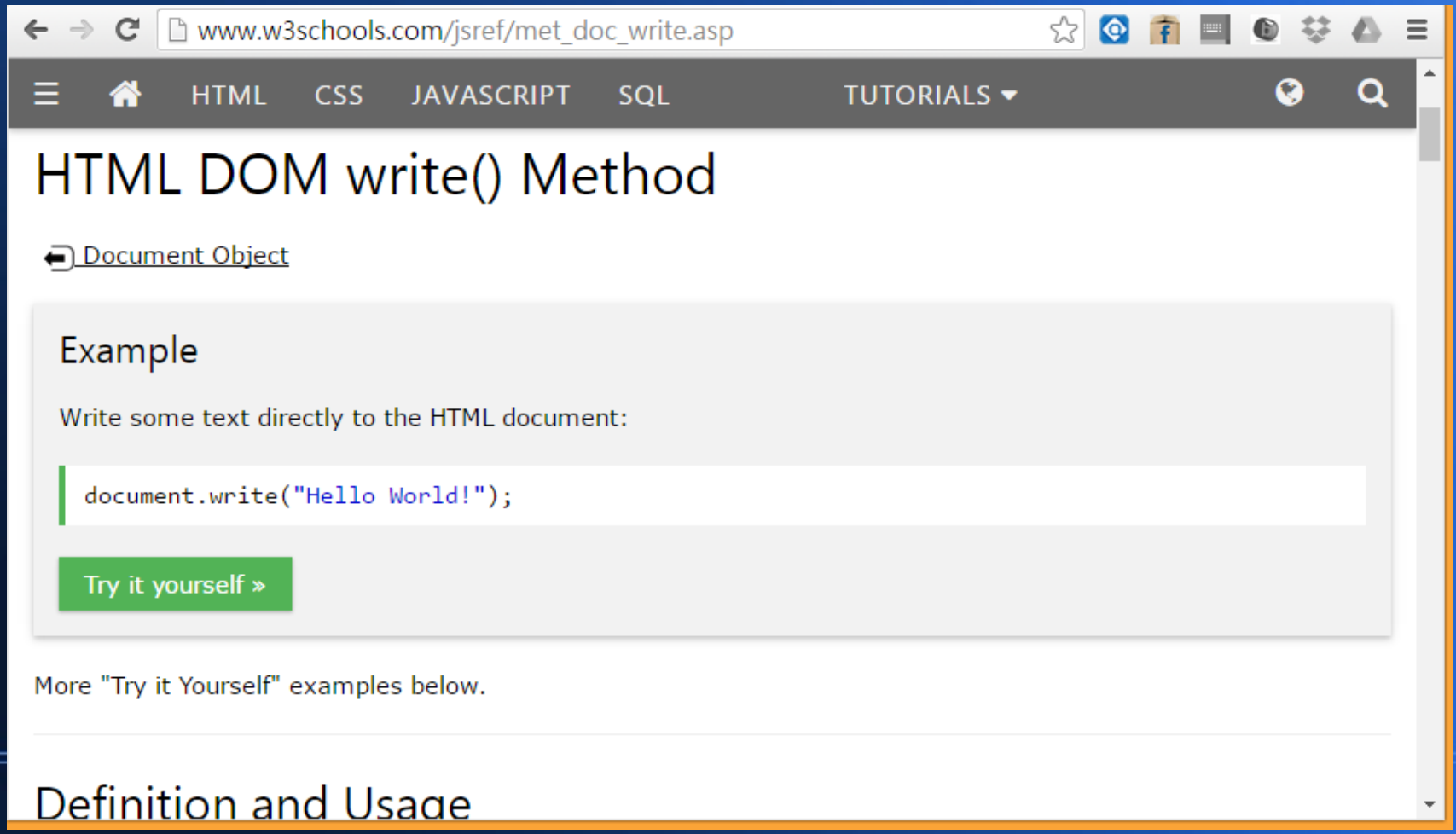
而您剛剛所引入的 .js 套件

- 也可以在 console 中呼叫並使用

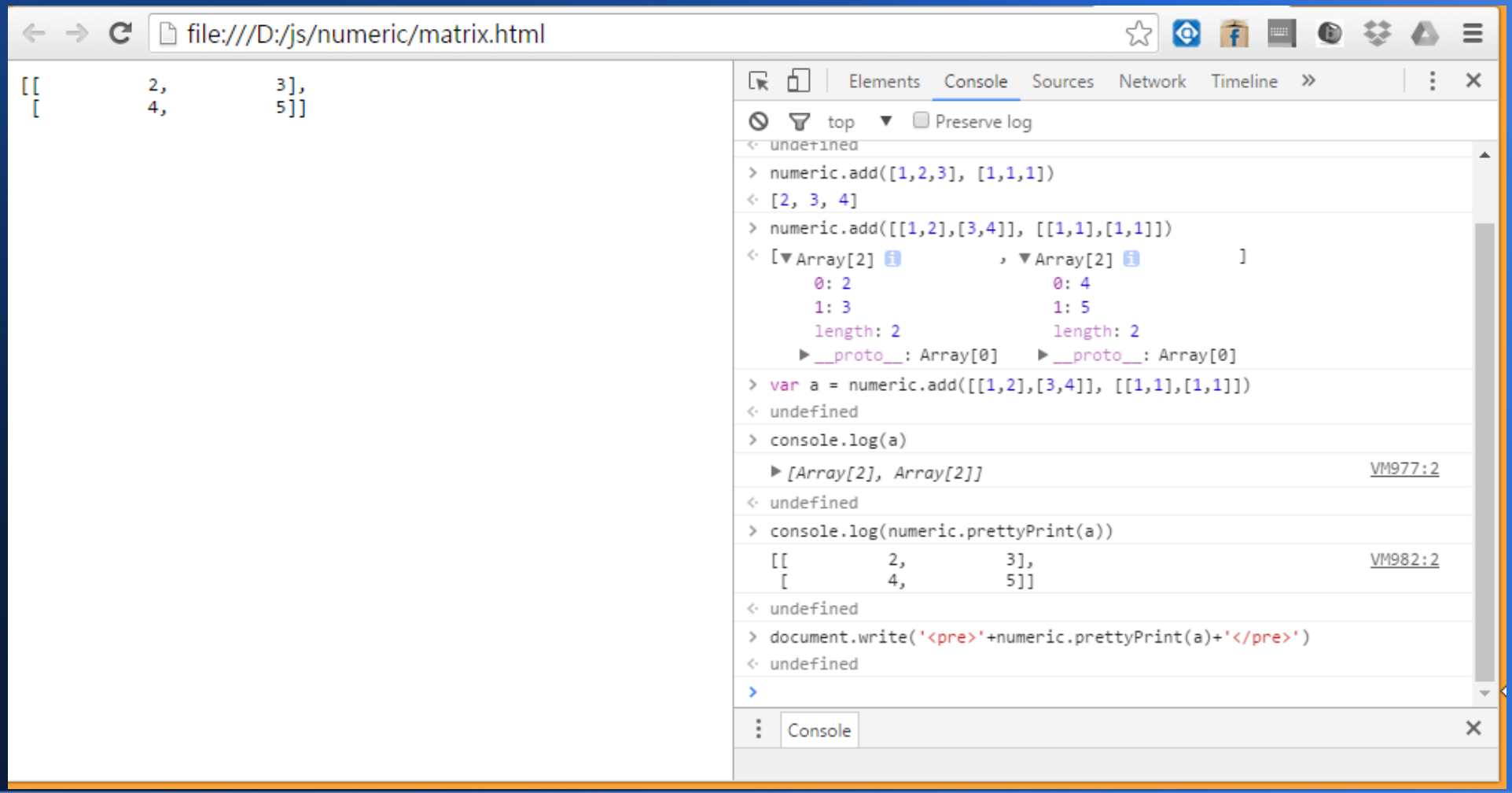


甚至如果你把結果輸出到網頁中

- 例如用 document.write(...)



那麼也會看到結果


file:///D:/js/numeric/matrix.html
[[2, 3],
[4, 5]]
Elements
Console
Sources
Network
Timeline
top
Preserve log
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
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]]
VM977:2
undefined
console.log(numeric.prettyPrint(a))
[[2, 3],
[4, 5]]
VM982:2
undefined
document.write('<pre>'+numeric.prettyPrint(a)+'</pre>')
undefined
Console

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

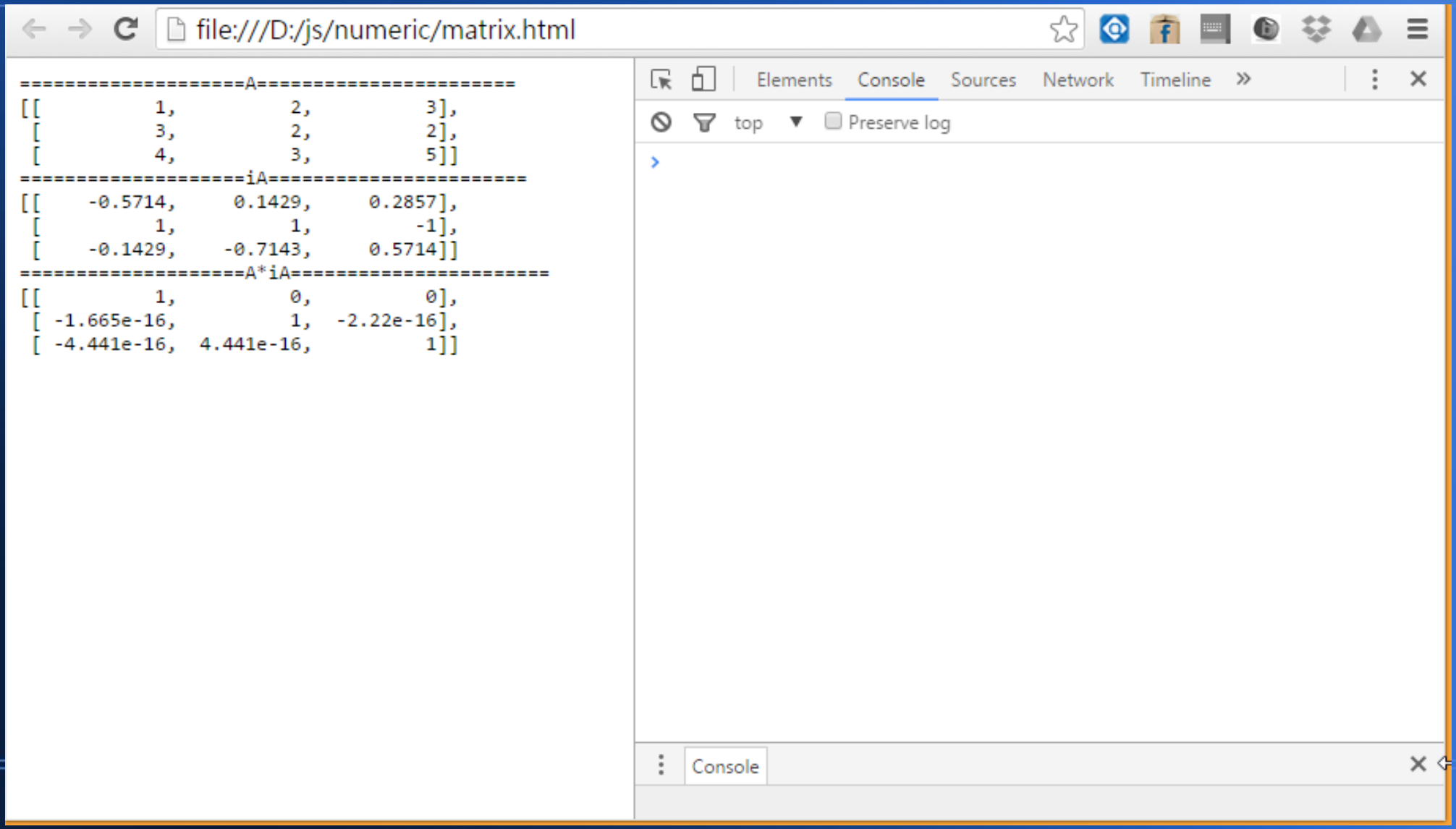
- 要使用該套件，可以直接將程式碼寫在網頁內！

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



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

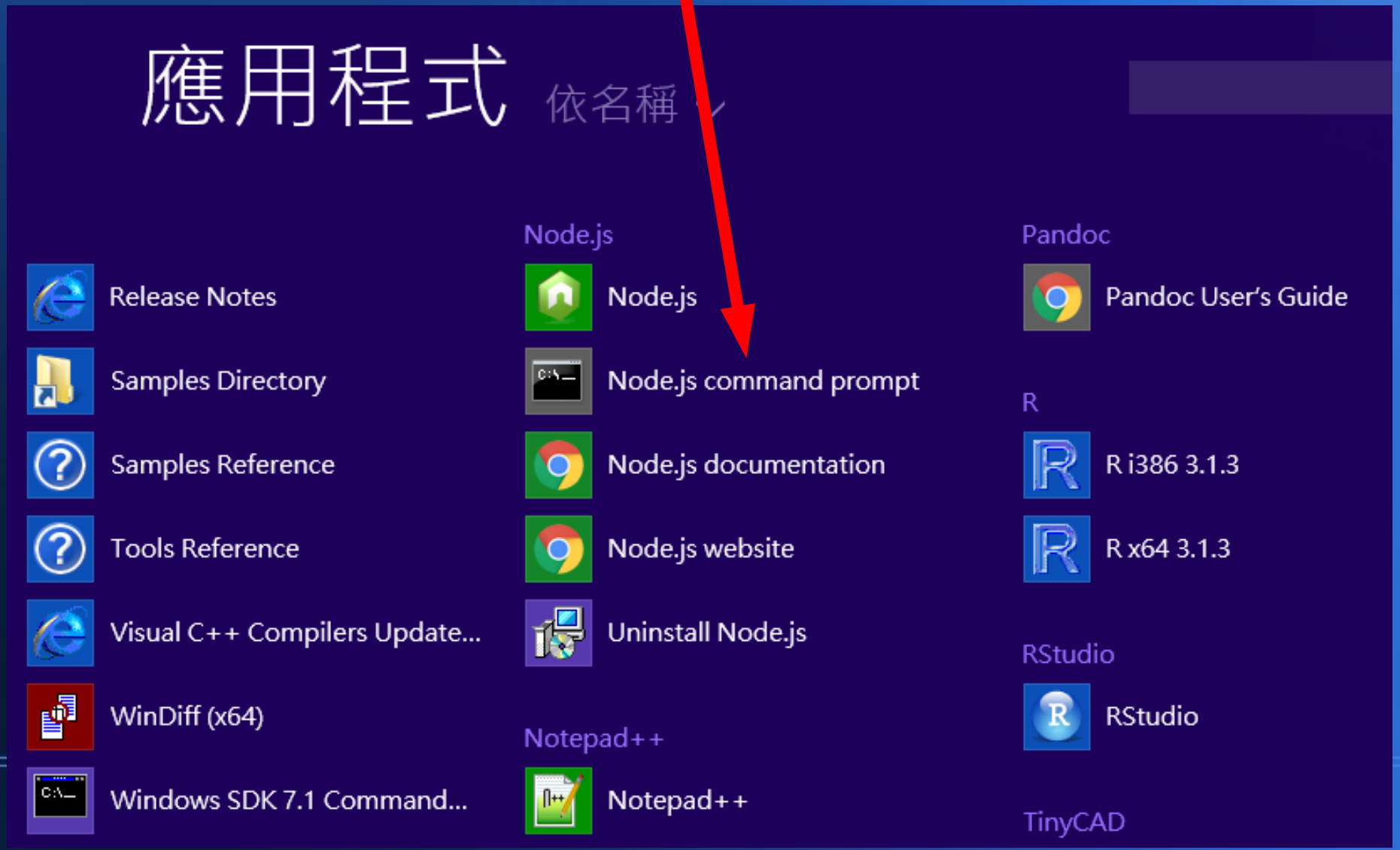
當然得先安裝 node. js

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



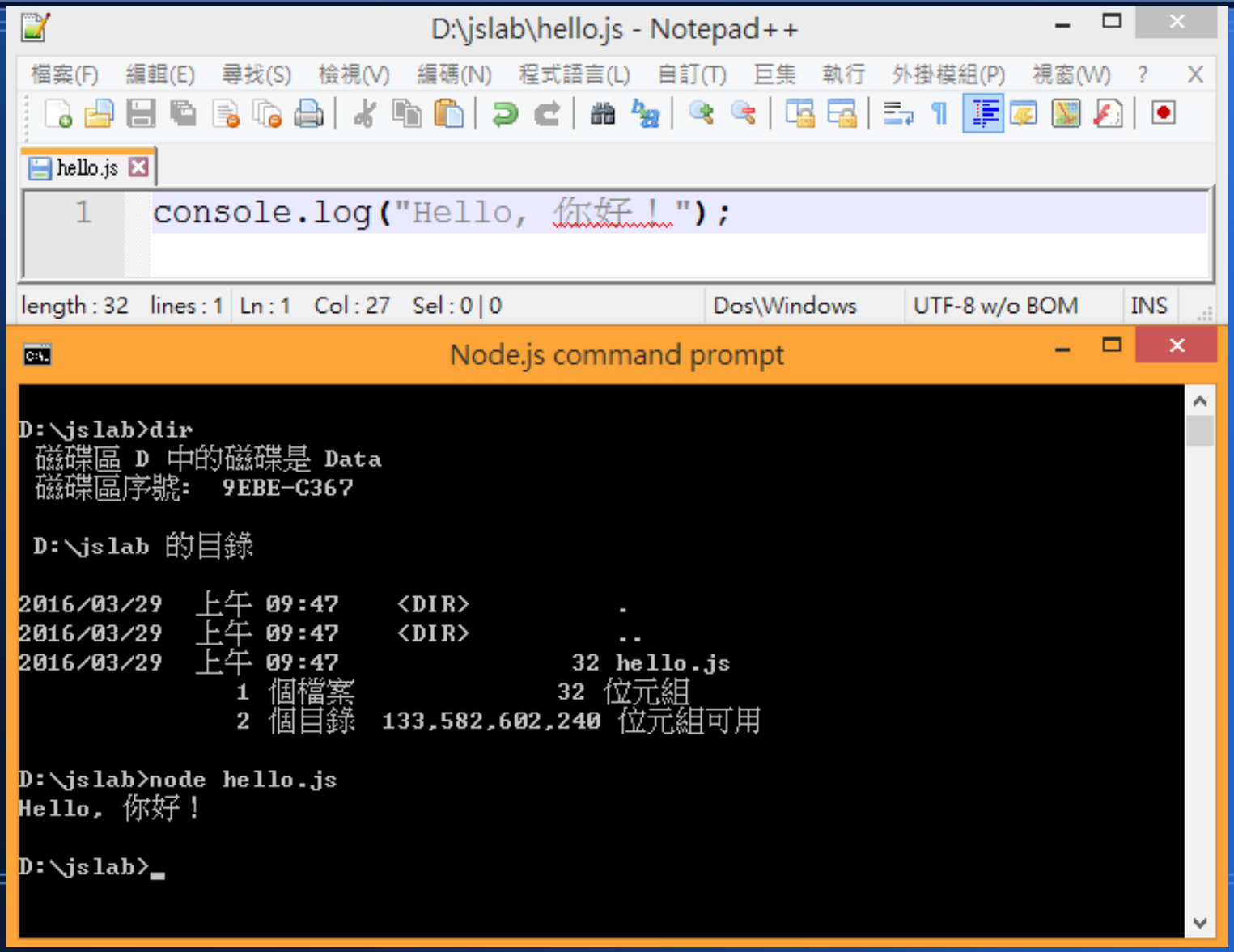
對於使用 MAC 或 Linux 的人

- 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>

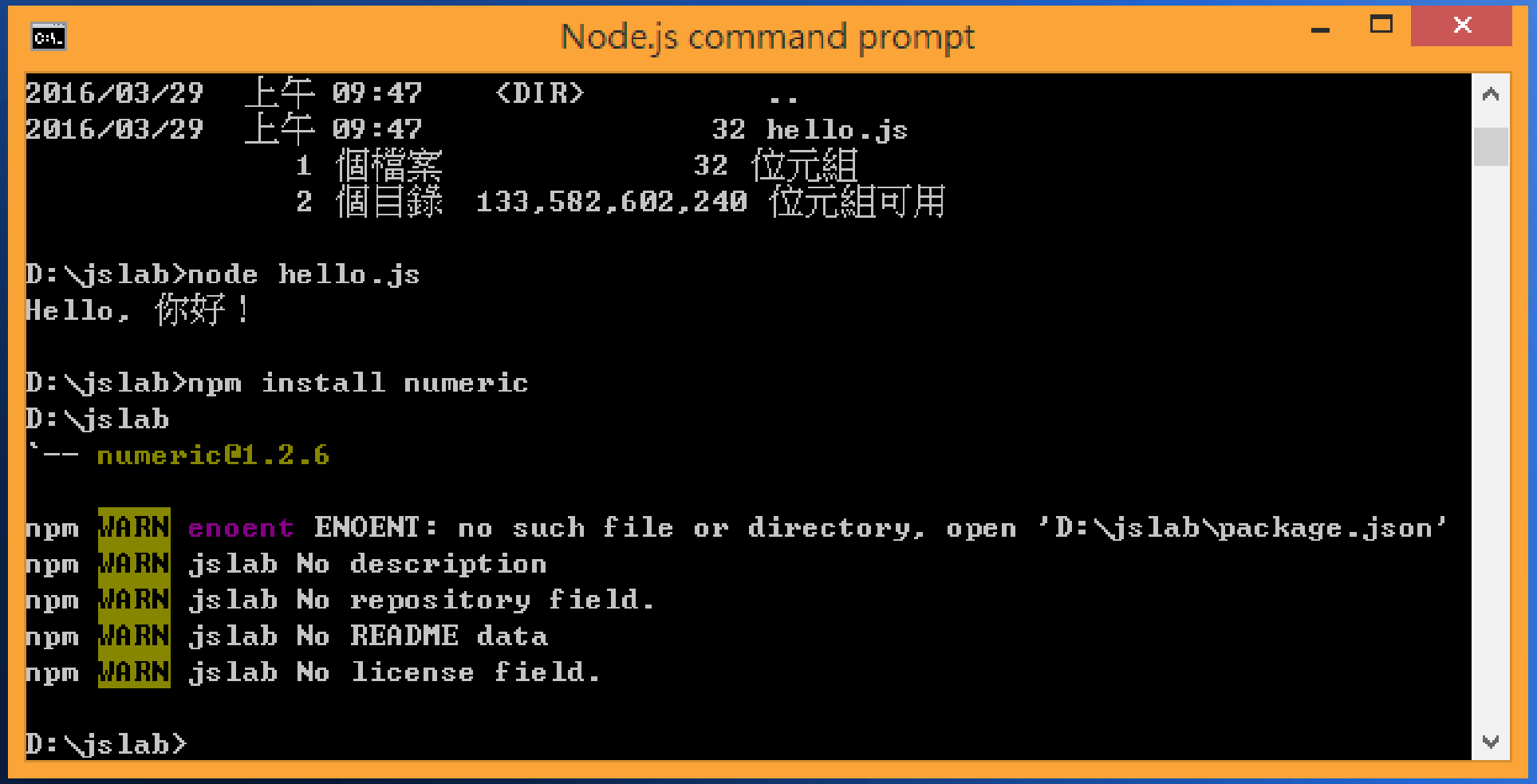
如果這些小程式可以正常執行

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

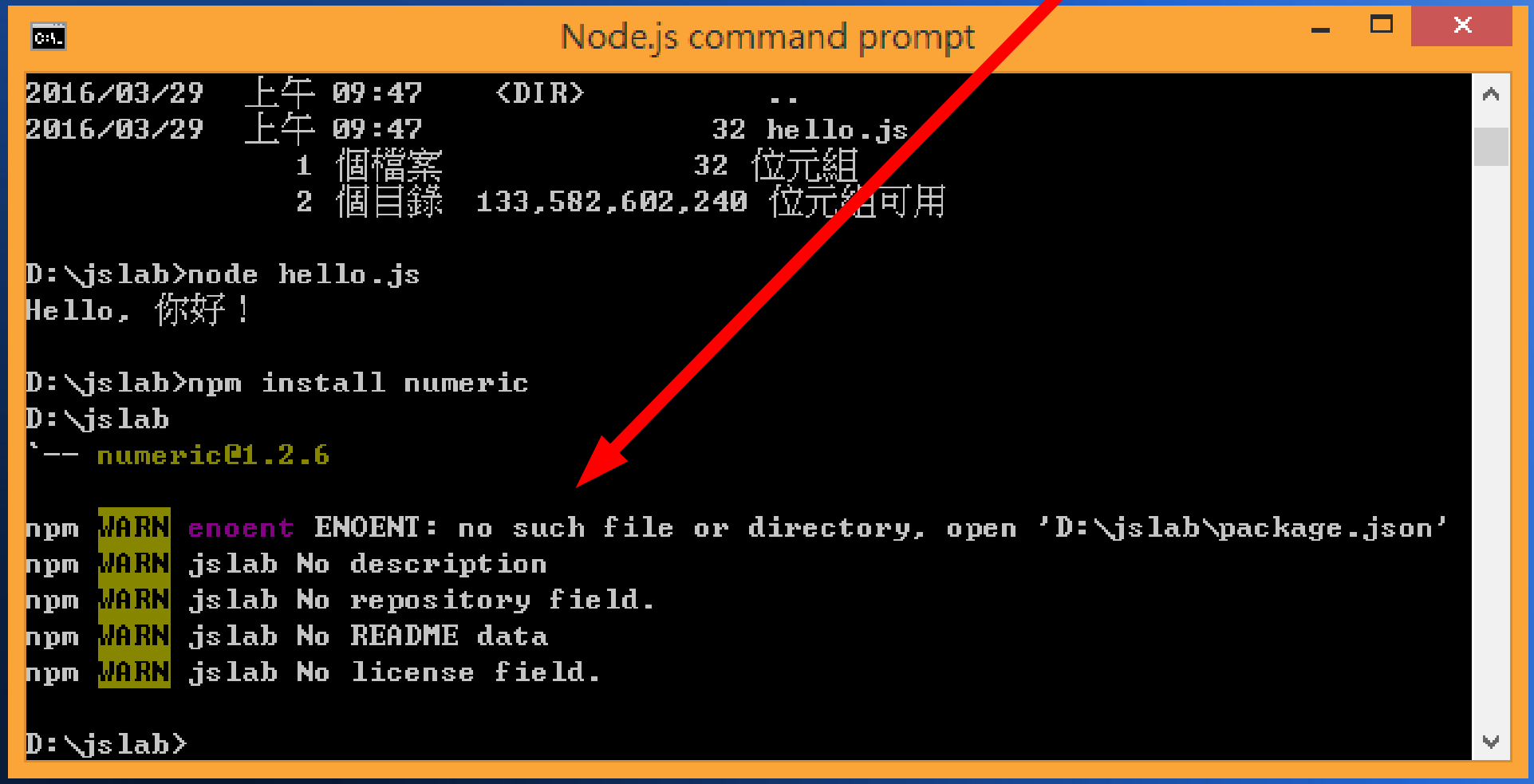
接著您可以開始安裝套件

- 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？



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

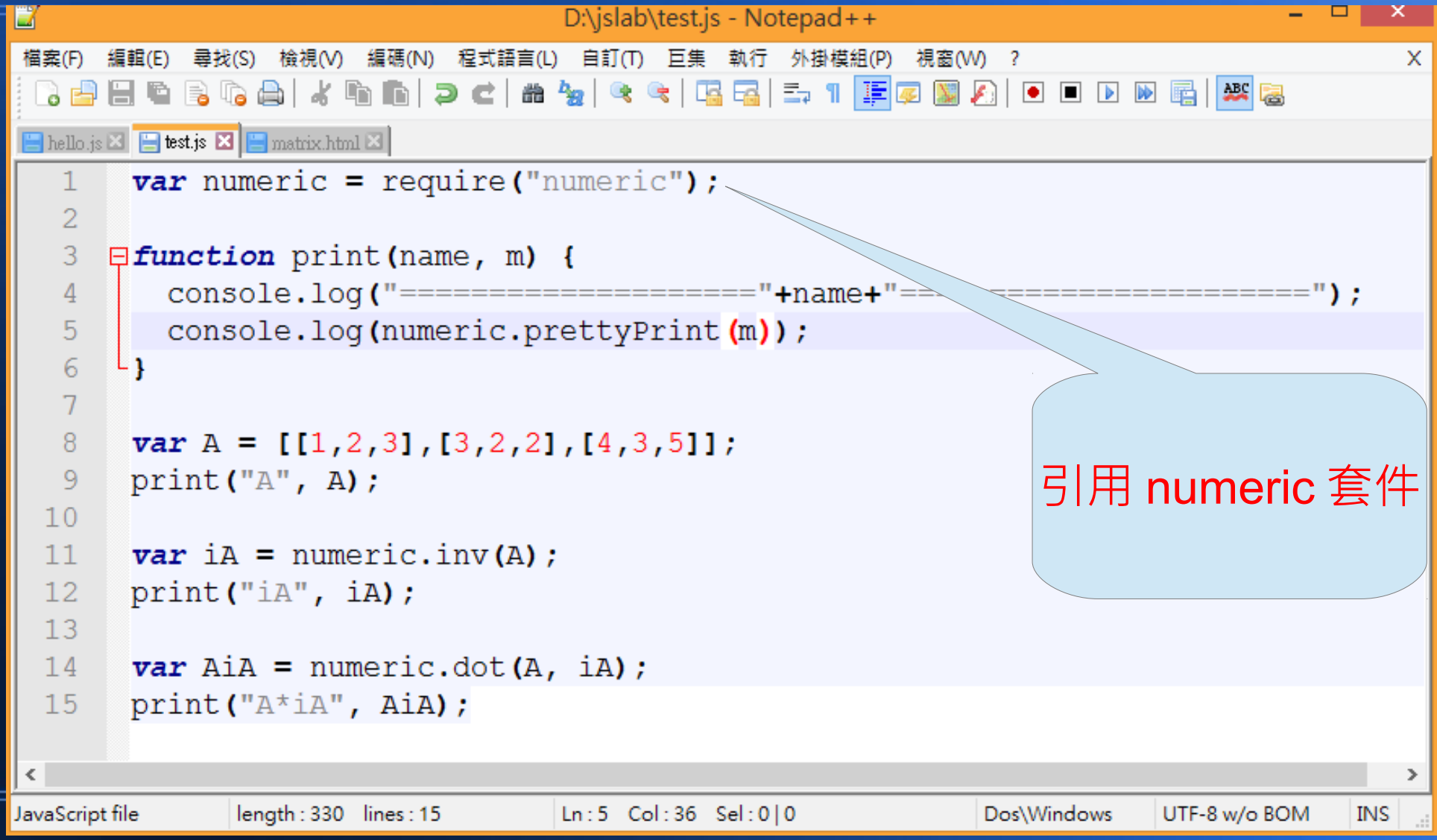
- 那些就是剛剛安裝的套件內容！

現在、讓我們開始在 node.js 裏使用

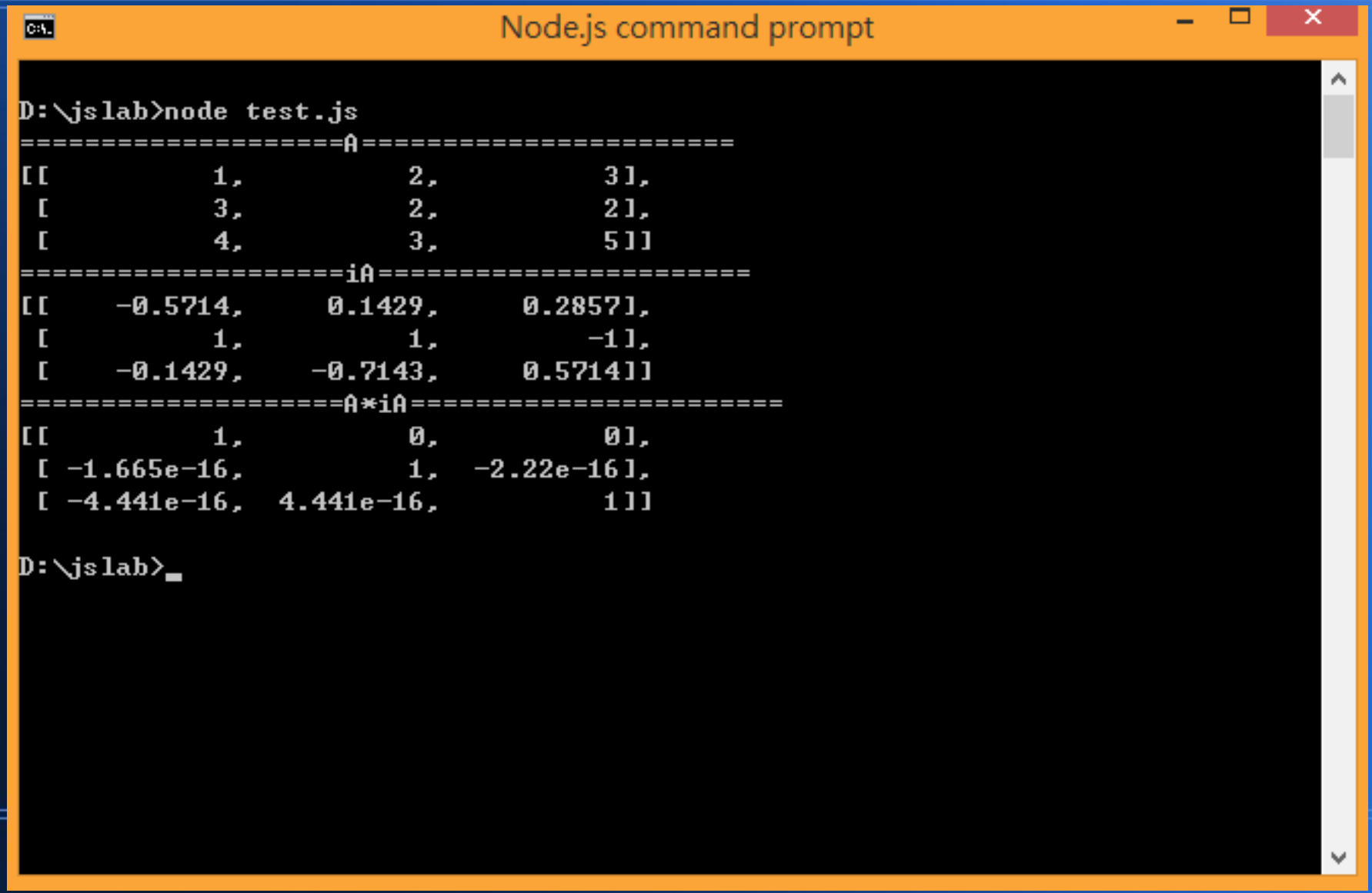
- 剛剛所安裝的 numeric 套件吧！

我們先寫一個測試程式

- 就叫 test. js 好了！



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？

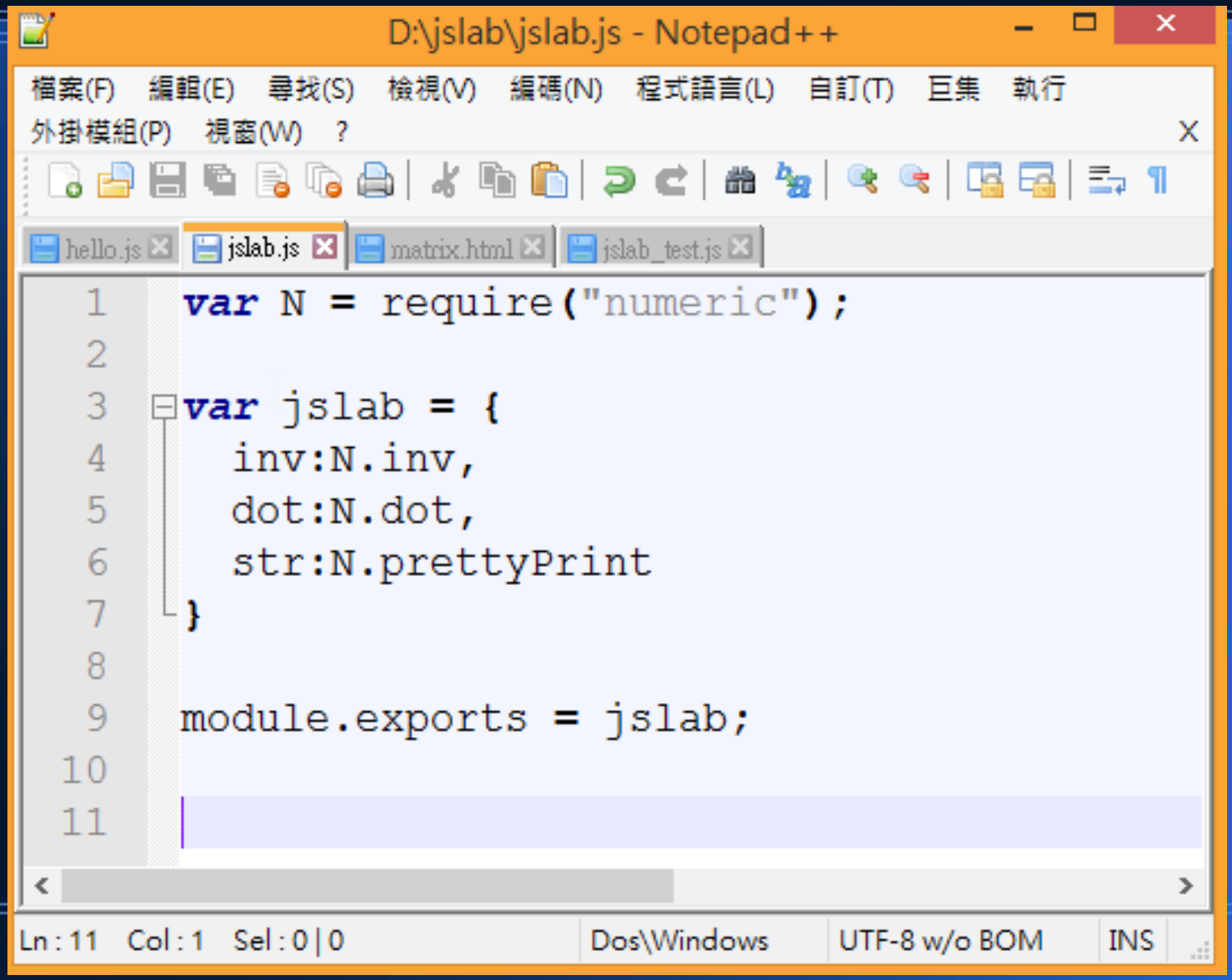
方法不難

- 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



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

hello.js | jslab.js | matrix.html | jslab_test.js

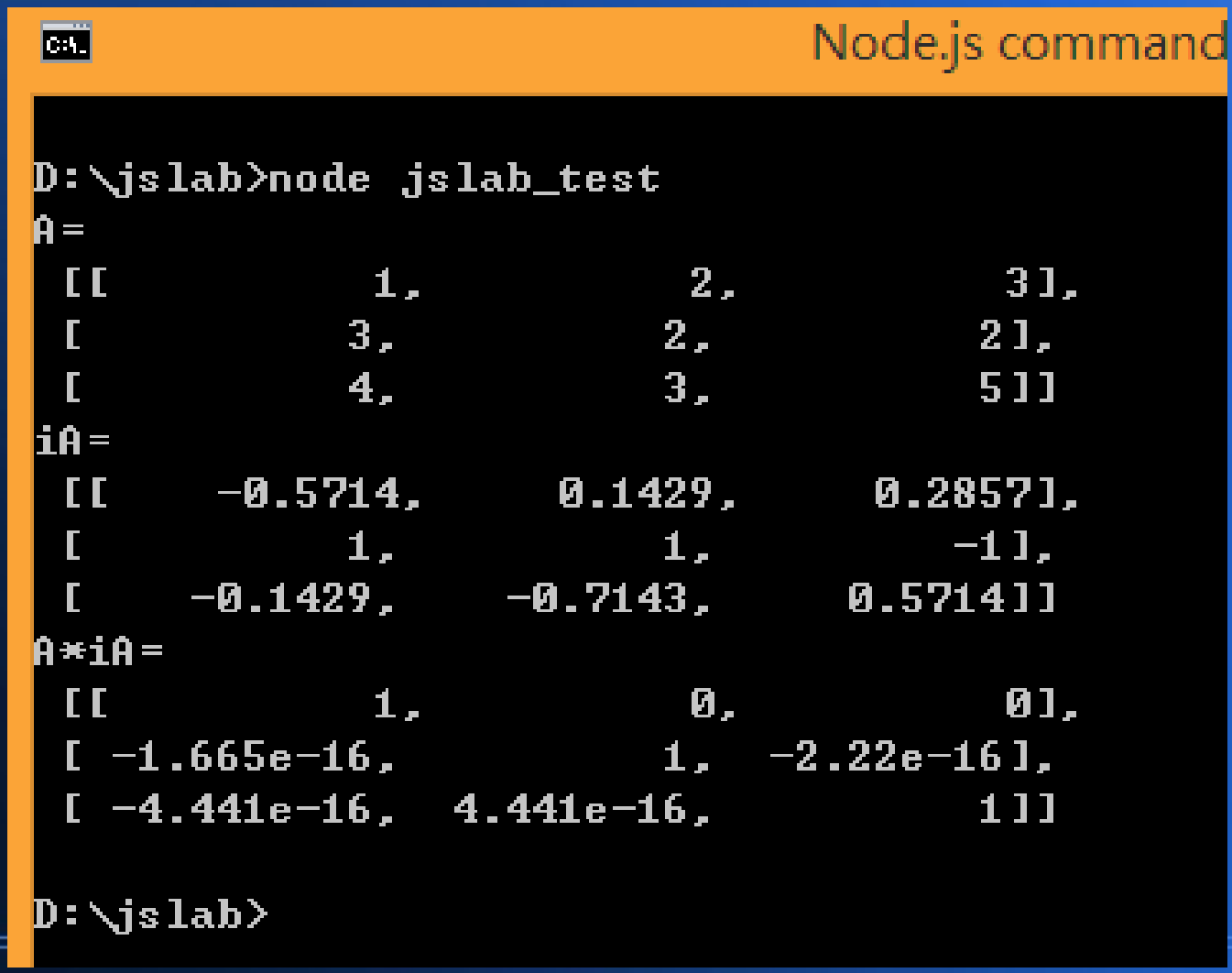
```
1  var N = require("numeric");
2
3  var jslab = {
4     inv:N.inv,
5     dot:N.dot,
6     str:N.prettyPrint
7  }
8
9  module.exports = jslab;
10
```

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

hello.js | jslab.js | matrix.html | jslab_test.js

```
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



換句話說、每個程式
只要用 module.exports 匯出物件

hello.js | jslab.js | matrix.html | jslab_test.js

```
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;
```

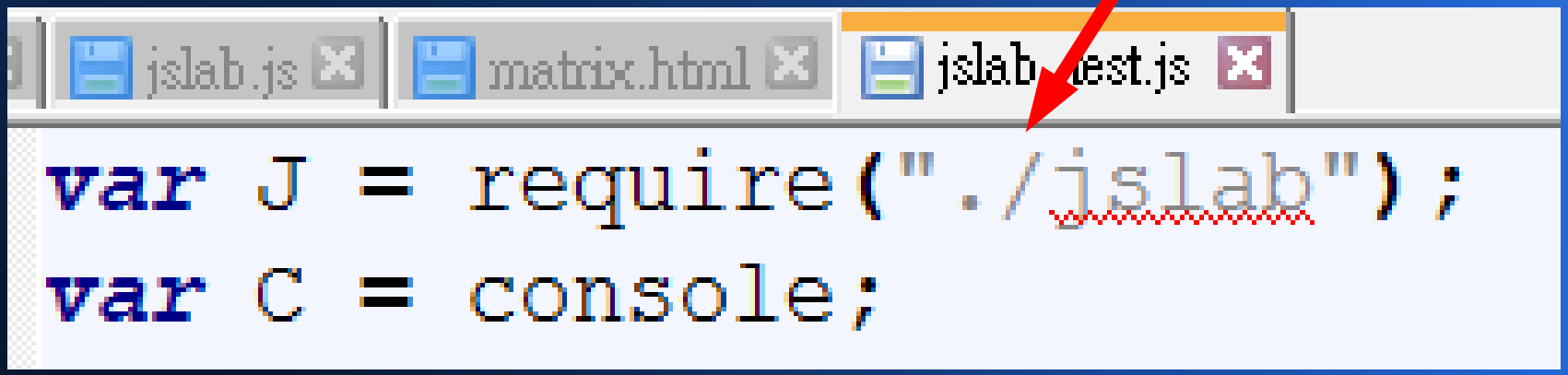
接著就可以用 require() 函數引入該物件來使用

hello.js | jslab.js | matrix.html | jslab_test.js

```
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));
```

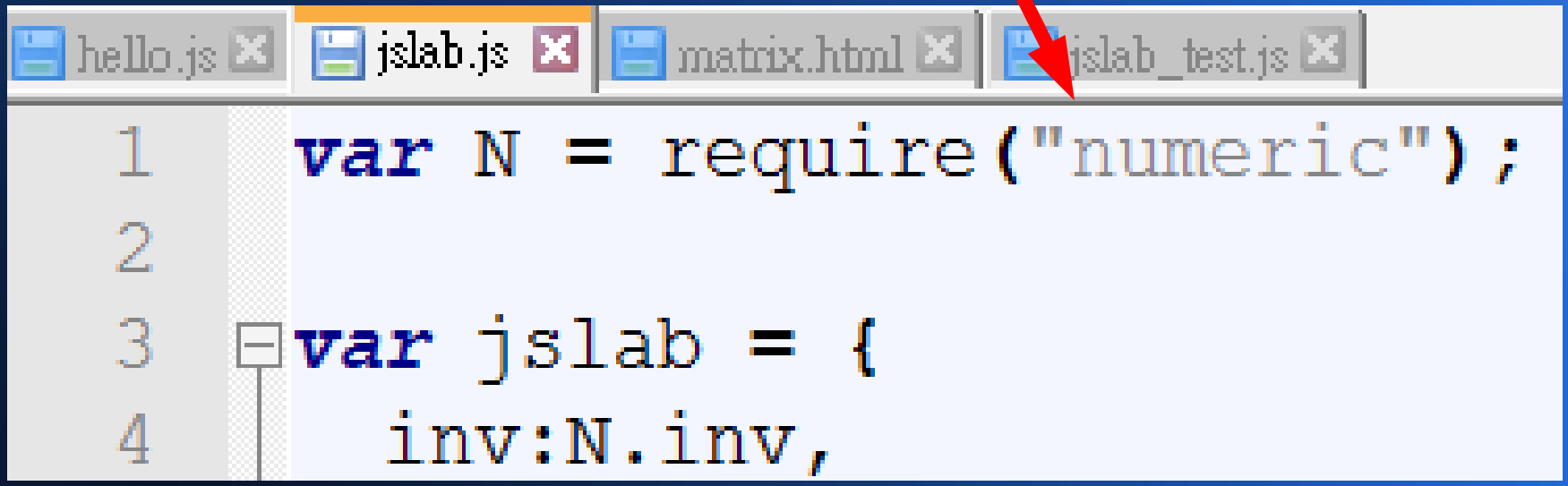
但必須注意的是

- 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。



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。



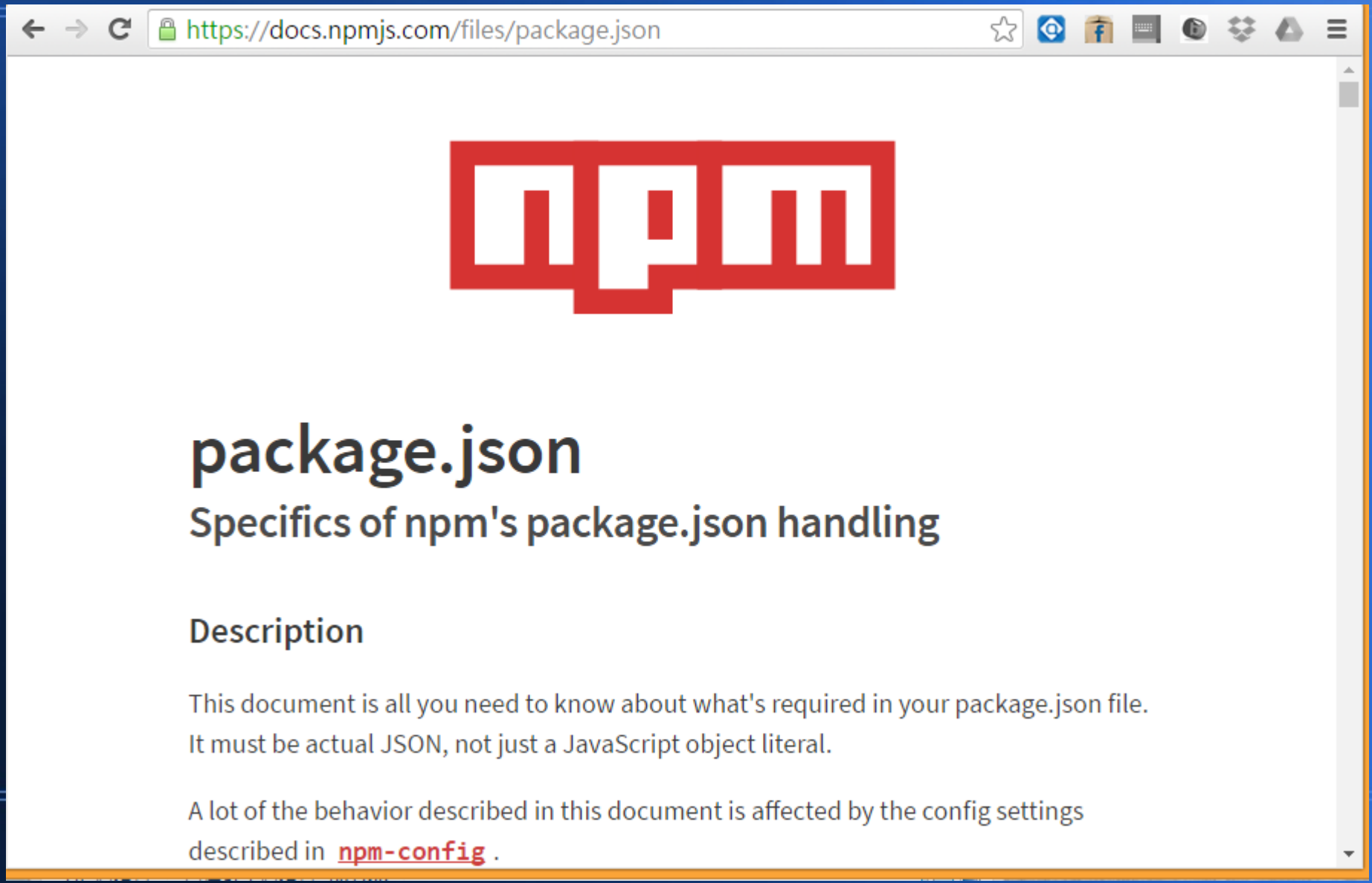
好了、現在我們已經學會了

- 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？

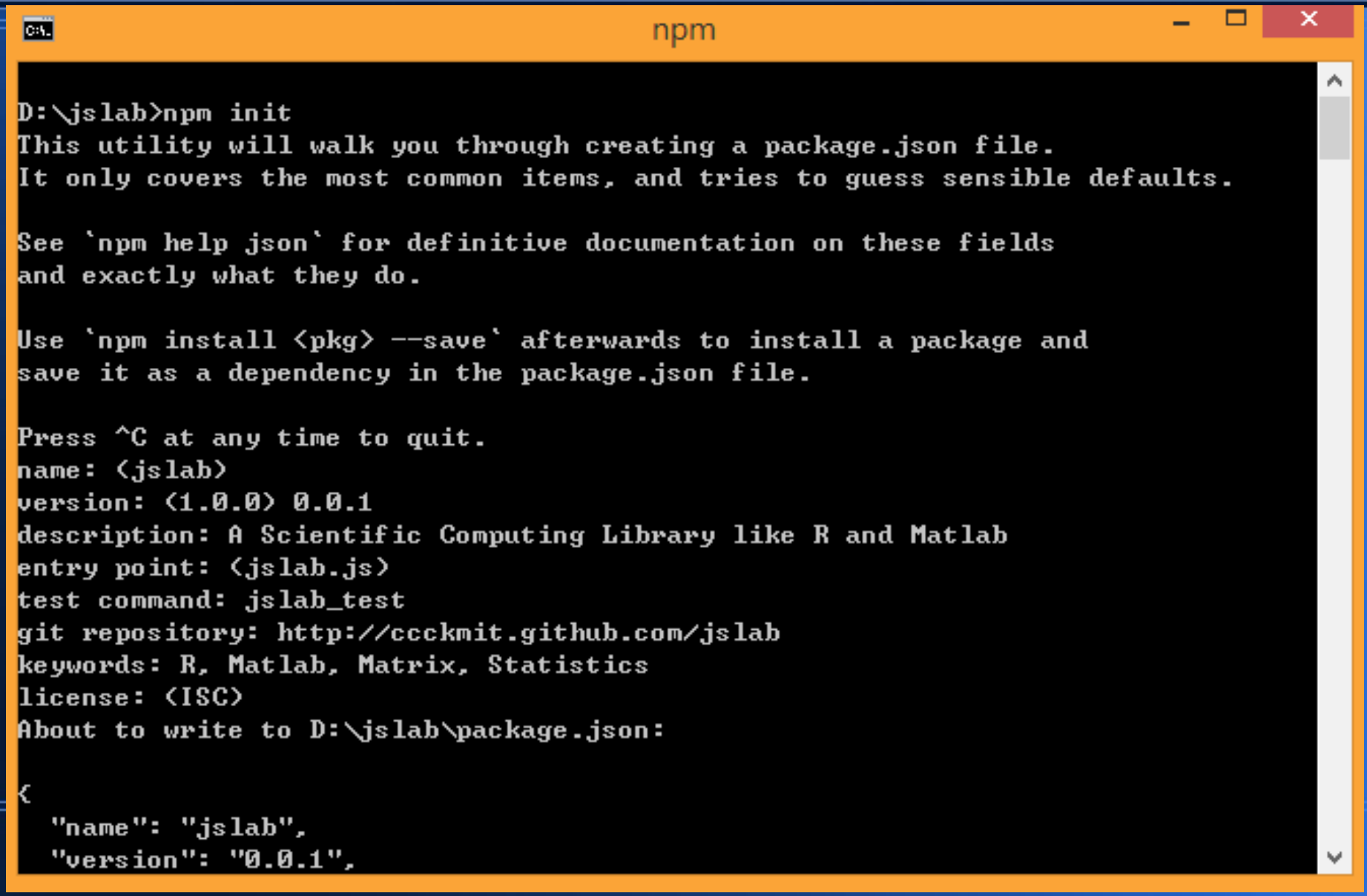
關於這個問題，就必須要請 package. json 登場了



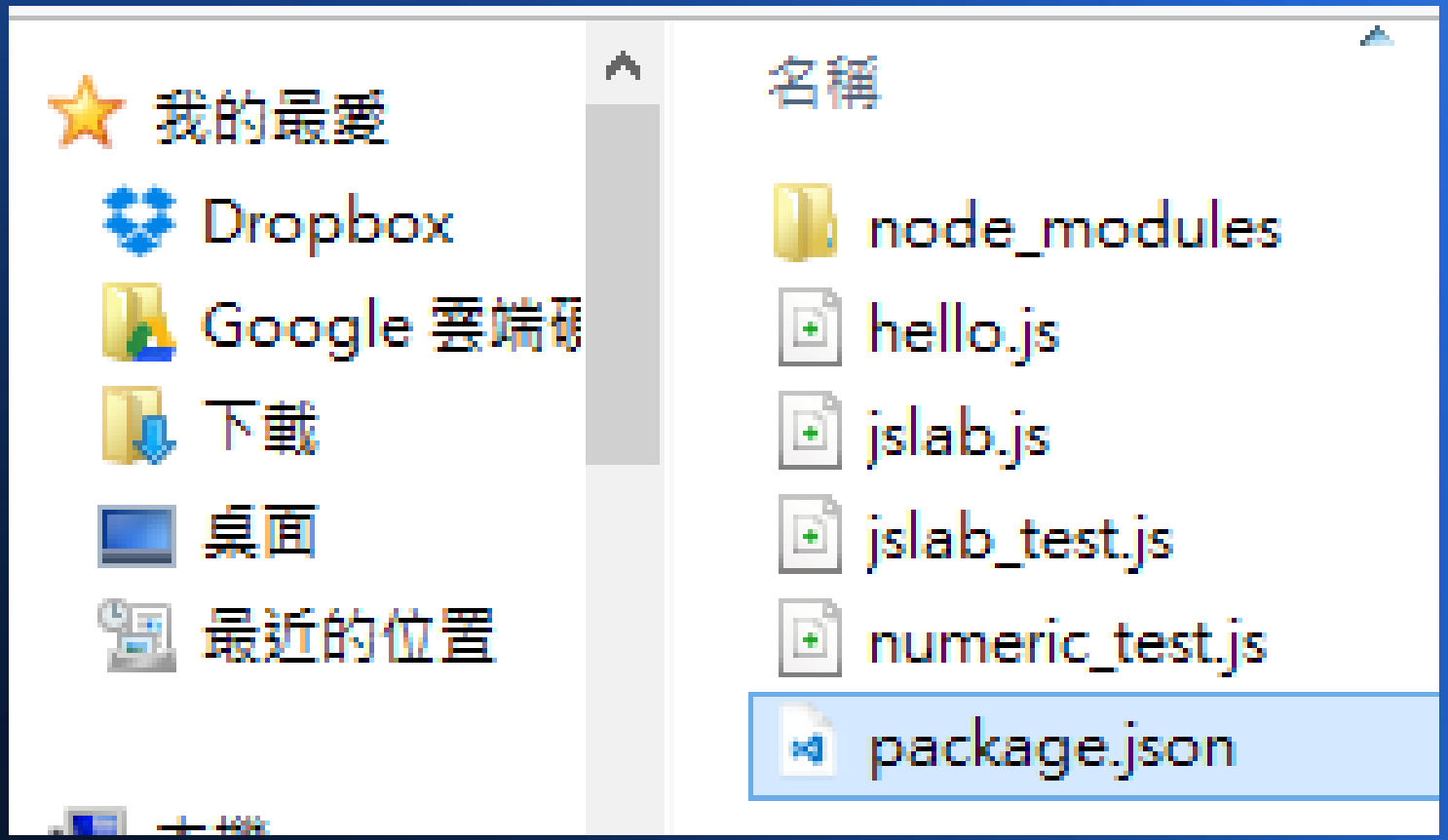
package. json

- 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



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

hello.js | jslab.js | matrix.html | jslab_test.js | package.json

```
{
  "name": "jslab",
  "version": "0.0.1",
  "description": "A Scientific Computing Library li
  "main": "jslab.js",
  "dependencies": {
    "numeric": "^1.2.6"
  },
  "devDependencies": {},
  "scripts": {
    "test": "jslab_test"
  },
  "repository": {
    "type": "git",
    "url": "http://ccckmit.github.com/jslab"
  },
```

這樣、我們就已經準備好

- 可以將 jslab 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 jslab 套件發佈到 npm 官網上

但是、在發佈之前

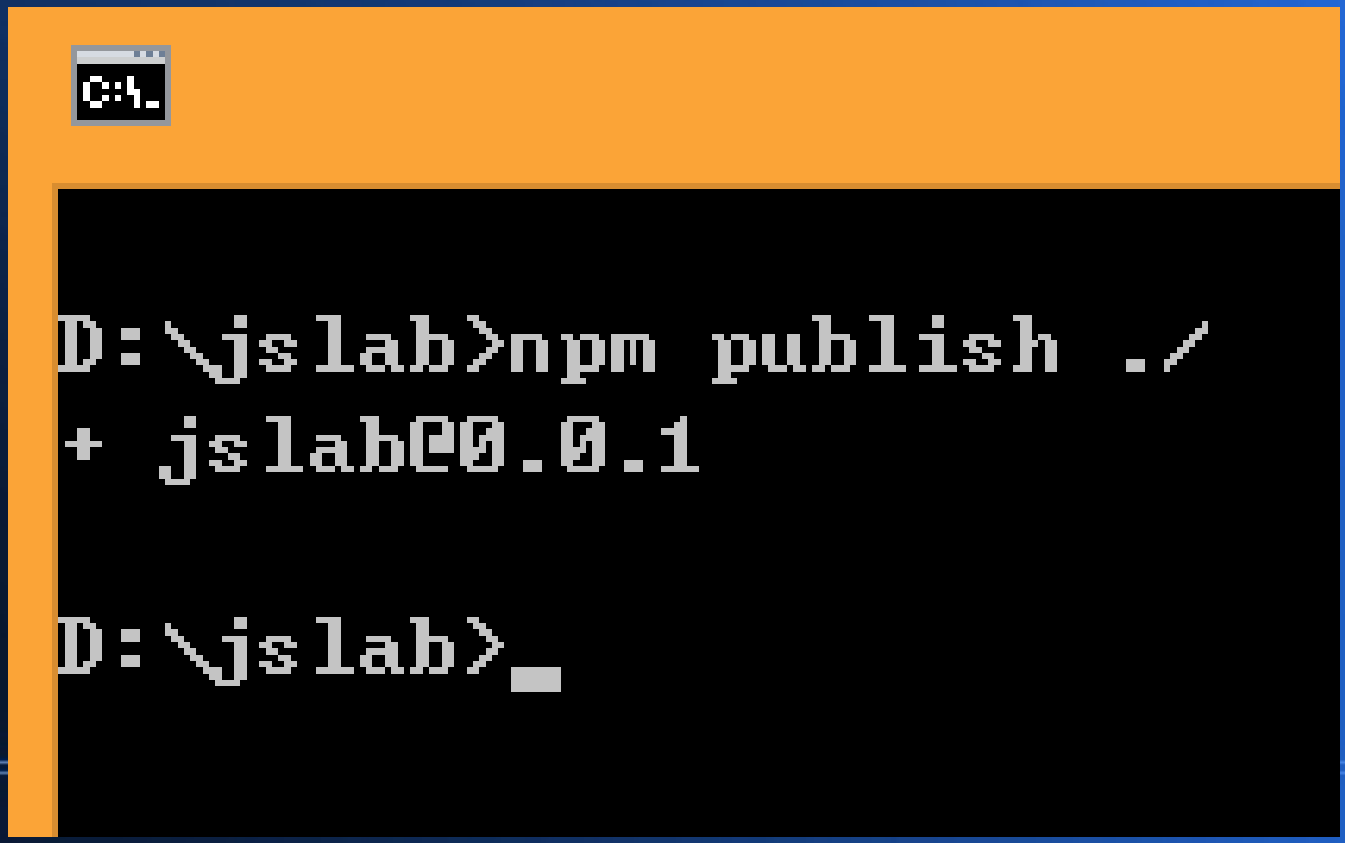
- 你必須先讓 npm 知道你是誰！

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

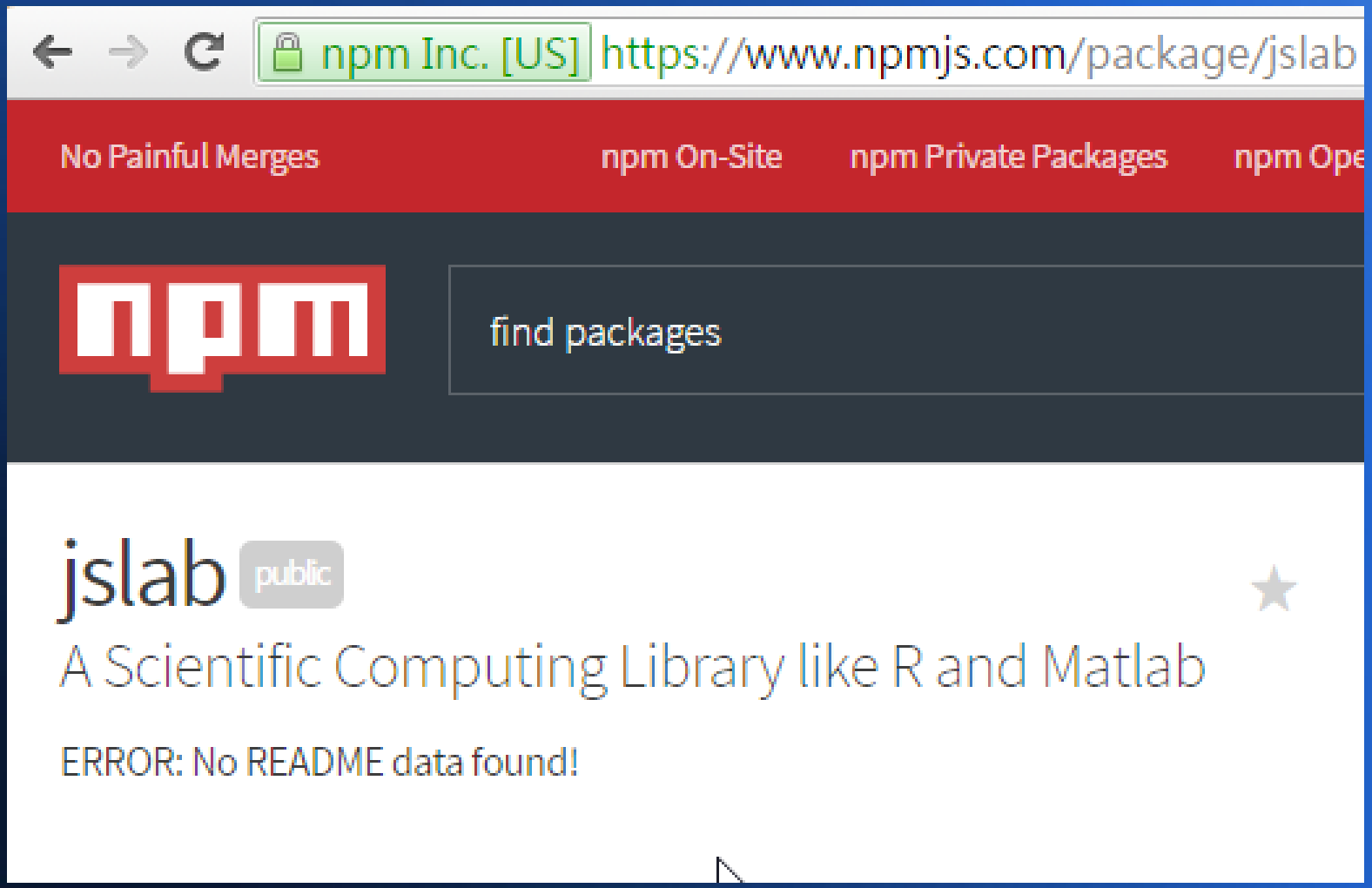
- npm publish ./



接著再上 npm 官網查一下

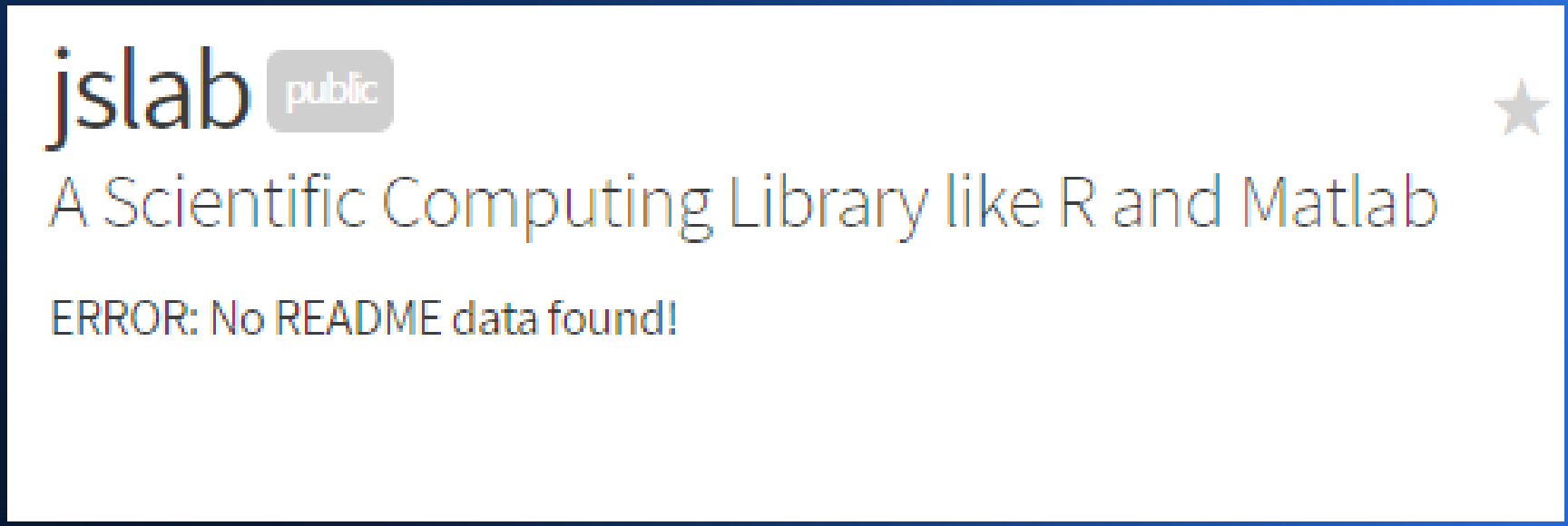
- 看看您的專案是否真的發布成功了

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



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

hello.js | jslab.js | matrix.html | jslab_test.js | package.json | README.md

````
jslab -- A Scientific Computing Library like R and Matlab

## Introduction

```
npm install jslab
```

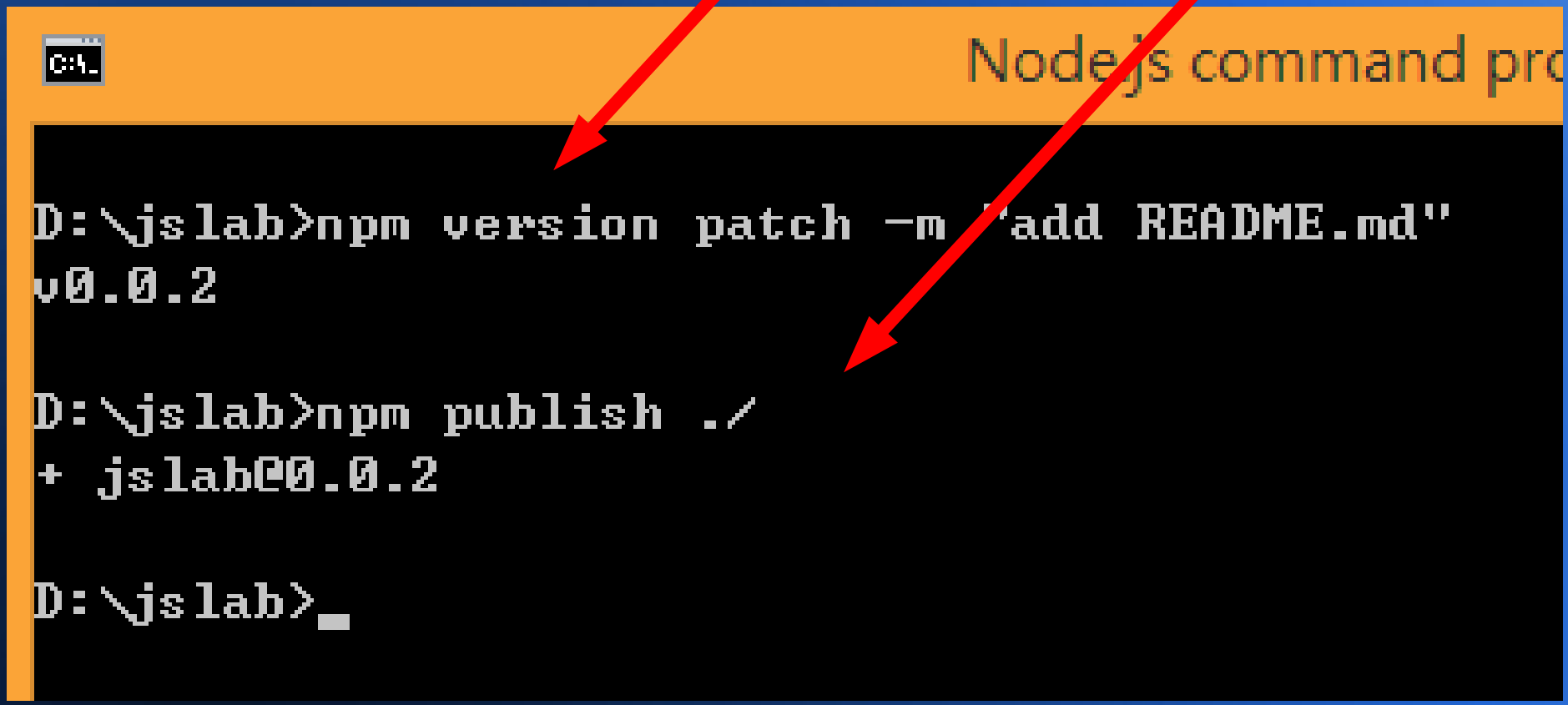
## Use jslab

file : jslab_test.js

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

var A = [[1,2,3],[3,2,2],[4,3,5]];
````
````

寫好後就可以更新版本再發佈



再檢查一下就會發現
README 資訊已經放上去了

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 官網上了！

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, postgreSQL 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 !