

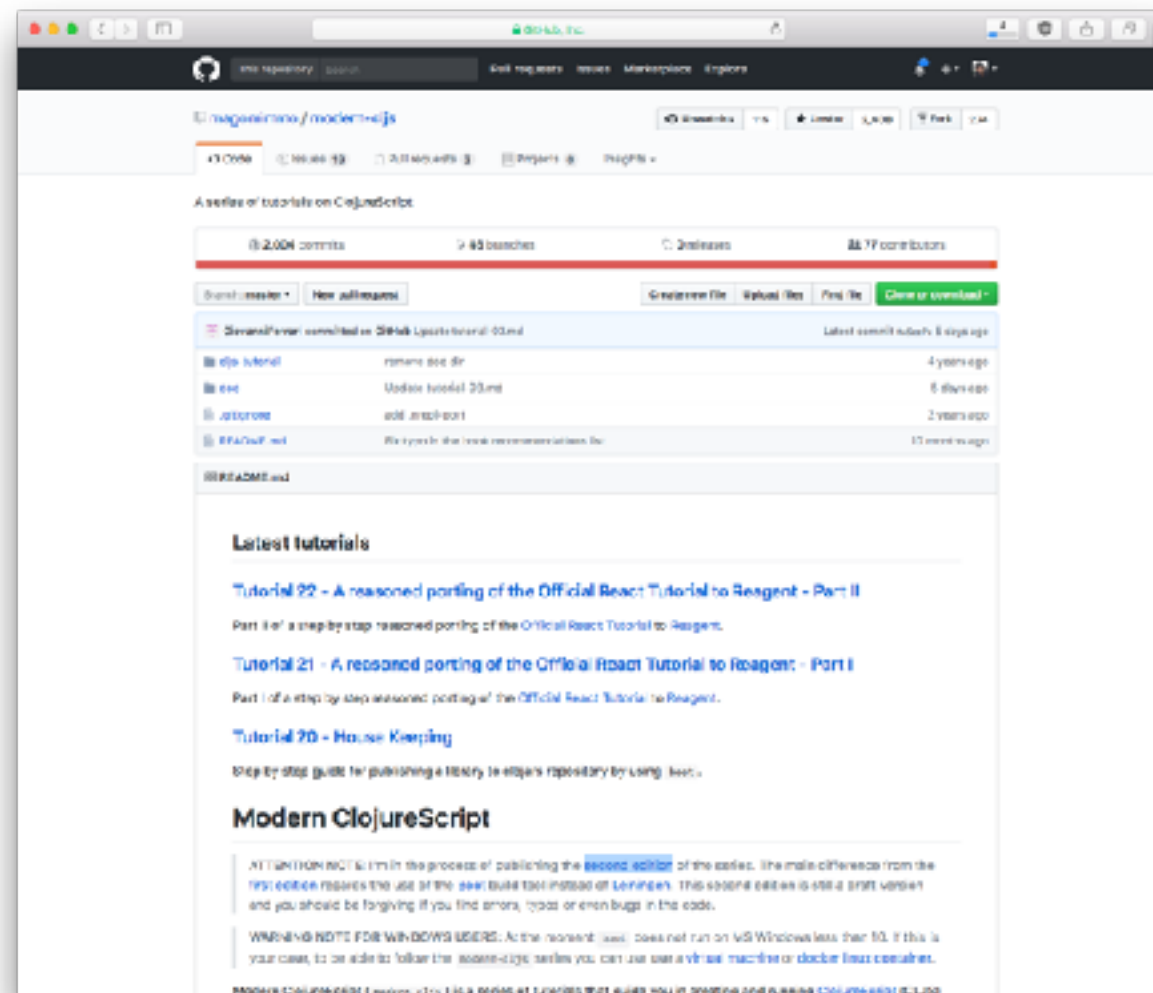
Modern CLJS 讀書會

(ch1 ~ ch2)

Coldnew

課程網址

- <https://github.com/magomimmo/modern-cljs>
- 目前有兩個版本: 選擇 second edition

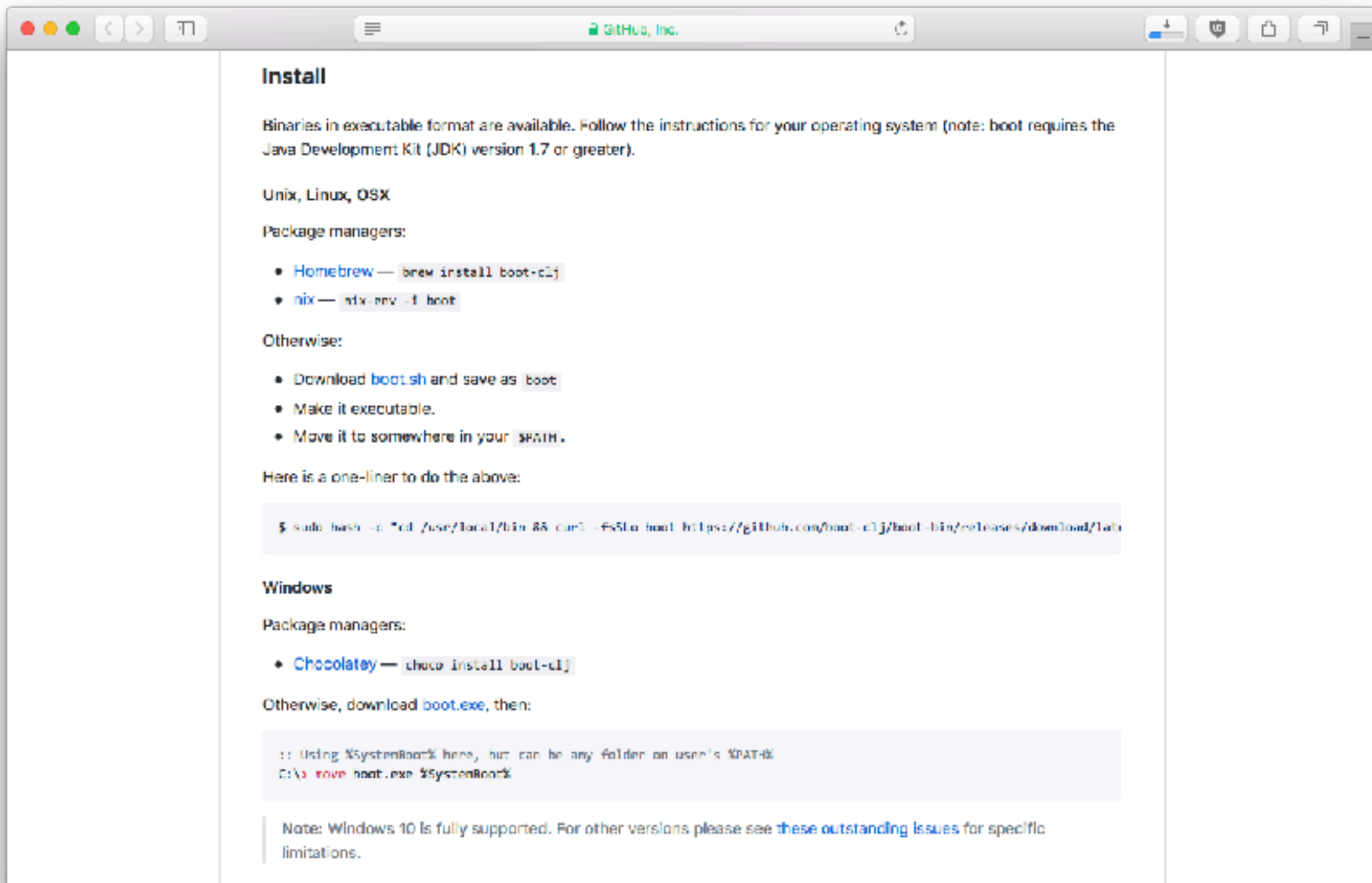


事前準備

- 安裝 boot
- 編輯器或是 IDE
 - Emacs (Spacemacs)
 - Cursive
 - Vim
- 其他: From JavaScript to ClojureScript: Which editor to choose?

安裝 boot

- 請參考: [Get Started](#)



Tutorial 1

The Basics

(second edition)

建立專案

```
2. coldnew@Fevia:~/modern-cljs (bash)
coldnew@Fevia ~/modern-cljs $ tree -L 4.
.
├── build.boot
├── html
│   └── index.html
└── src
    └── cljs
        └── modern_cljs
            └── core.cljs

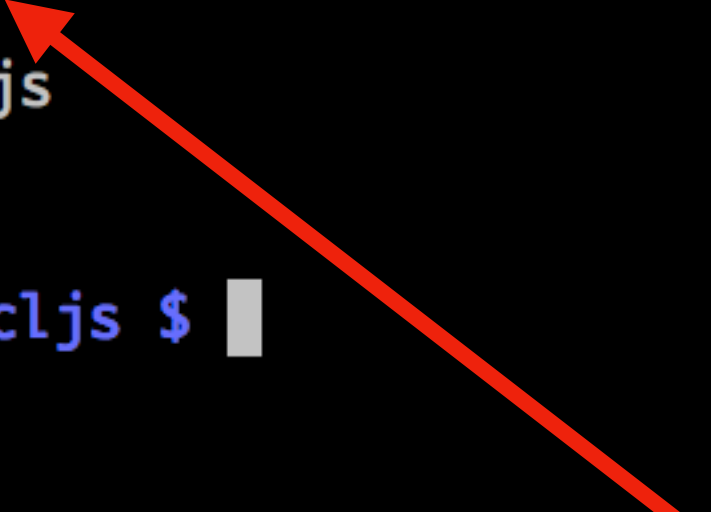
4 directories, 3 files
coldnew@Fevia ~/modern-cljs $
```

我們手動建立這樣的專案結構
(這步驟可以透過 `lein new` 以及樣板產生)

建立專案

```
2. coldnew@Fevia:~/modern-cljs (bash)
coldnew@Fevia ~/modern-cljs $ tree -L 4.
.
├── build.boot
├── html
│   └── index.html
└── src
    └── cljs
        └── modern_cljs
            └── core.cljs

4 directories, 3 files
coldnew@Fevia ~/modern-cljs $
```

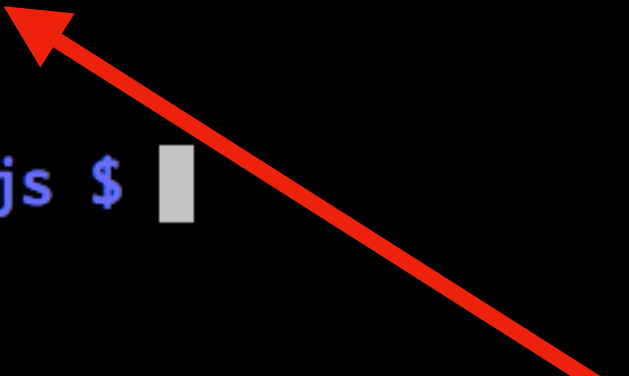


如果要建立 namespace 為 modern-cljs 的程式，資料夾要把 - 改成 _
(這是 Java 規範的限制)

建立專案

```
2. coldnew@Fevia:~/modern-cljs (bash)
coldnew@Fevia ~/modern-cljs $ tree -L 4.
.
├── build.boot
├── html
│   └── index.html
└── src
    └── cljs
        └── modern_cljs
            └── core.cljs

4 directories, 3 files
coldnew@Fevia ~/modern-cljs $
```



ClojureScript 的副檔名為 **.cljs**

Hello World ! (core.cljs)

```
;; create the main project namespace
```

```
(ns modern-cljs.core) ——— 這個檔案的命名空間
```

```
;; enable cljs to print to the JS console of the browser  
(enable-console-print!)
```

在 ClojureScript 中，如果想使用 println 之壘的輸出，要加上上面這個設定

```
;; print to the console  
(println "Hello, World!")
```

這邊就和 js 的 console.log("Hello, World!"); 同義

編輯 `src/cljs/modern_cljs/core.cljs` 加入以上程式碼

Hello World ! (index.html)

```
<!doctype html>
<html>
  <head>
    <title>Hello, World!</title>
  </head>
  <body>
    <script src="main.js"></script>
  </body>
</html>
```

編譯好的 cljs 位置



編輯 [html/index.html](#) 加入以上程式碼

Hello World ! (build.boot)

設定 boot 的參數

ClojureScript 檔案位置

```
(set-env!  
  :source-paths #{ "src/cljs" }  
  :resource-paths #{ "html" }  
  
  :dependencies '[[adzerk/boot-cljs "1.7.228-2"]])  
  
(require '[adzerk.boot-cljs :refer [cljs]])
```

Html 或其他非 cljs 檔案位置
(包含圖片、css ...etc)

這個專案會用到的依賴函式庫

載入和 cljs 相關的 task

編輯 **build.boot** 加入以上程式碼, boot 在 ClojureScript 下是類似 npm、yarn 之類的角色

boot -h

```
boot -h

...

Tasks:  ...
        target      Writes output files to the given dir...
        ...
        zip          Build a zip file for the project.
        cljs         Compile ClojureScript applications.

...

Do `boot <task> -h` to see usage info and TASK_OPTS for <task>.
```

使用 **boot -h** 可以看到 boot 的使用方式

boot cljs -h

```
2. coldnew@Fevia:~/modern-cljs (bash)
coldnew@Fevia ~/modern-cljs $ boot cljs -h
Classpath conflict: org.clojure/clojure version 1.8.0 already loaded, NOT loading version 1.2.0
Compile ClojureScript applications.

Multiple builds can be compiled parallel. To define builds use .cljs.edn
files. ID of build is the name of .cljs.edn file without the extension.
To compile only specific builds, use ids option to select .cljs.edn files
by name. Output files of build will be put below id.out folder in fileset.

If no .cljs.edn files exists, default one is created. It will depend on
all .cljs files in fileset.

Available --optimization levels (default 'none'):

* none          No optimizations. Bypass the Closure compiler completely.
* whitespace    Remove comments, unnecessary whitespace, and punctuation.
* simple        Whitespace + local variable and function parameter renaming.
* advanced       Simple + aggressive renaming, inlining, dead code elimination.

Source maps can be enabled via the --source-map flag. This provides what the
browser needs to map locations in the compiled JavaScript to the corresponding
locations in the original ClojureScript source files.

The --compiler-options option can be used to set any other options that should
be passed to the Clojurescript compiler. A full list of options can be found
here: https://github.com/clojure/clojurescript/wiki/Compiler-Options.
```

編譯專案

```
2. coldnew@Fevia:~/modern-cljs (bash)
coldnew@Fevia ~/modern-cljs $ boot cljs
Classpath conflict: org.clojure/clojure version 1.8.0 already loaded, NOT loading version 1.2.0
Writing main.cljs.edn...
Compiling ClojureScript...
• main.js
coldnew@Fevia ~/modern-cljs $ tree -L 4 .
.
├── build.boot
├── html
│   └── index.html
└── src
    ├── cljs
    │   ├── modern_cljs
    │   └── core.cljs
    └── main.js

4 directories, 3 files
coldnew@Fevia ~/modern-cljs $
```

用 boot cljs 編譯專案，
不過為何看不到編譯出來的 main.js 呢？

boot target task

```
2. coldnew@Fevia:~/modern-cljs (java)
coldnew@Fevia ~/modern-cljs $ boot target -h
Classpath conflict: org.clojure/clojure version 1.8.0 already loaded, NOT loading version 1.2.0
Writes output files to the given directory on the filesystem.

Options:
-h, --help      Print this help info.
-d, --dir PATH  Conj PATH onto the set of directories to write to (target).
-m, --mode VAL  VAL sets the mode of written files in 'rwxrwxrwx' format.
-L, --no-link   Don't create hard links.
-C, --no-clean  Don't clean target before writing project files.
coldnew@Fevia ~/modern-cljs $
```

**Boot 預設會將產生的結果存在系統的暫存空間 (ramdisk .. etc),
如果想要看到結果, 就需要 boot target 這個 task**

編譯專案 (boot target)

```
2. coldnew@Fevia:~/modern-cljs (bash)
coldnew@Fevia ~/modern-cljs $ boot cljs target
Classpath conflict: org.clojure/clojure version 1.8.0 already loaded, NOT loading version 1.2.0
Writing main.cljs.edn...
Compiling ClojureScript...
• main.js
Writing target dir(s)...
coldnew@Fevia ~/modern-cljs $ tree -L 3 .
.
├── build.boot
├── html
│   └── index.html
├── src
│   └── cljs
│       └── modern_cljs
└── target
    ├── index.html
    ├── main.js
    └── main.out
        ├── boot
        ├── cljs
        ├── cljs_deps.js
        ├── goog
        └── modern_cljs
```

編譯出來的結果都在 target 資料夾

boot.properties

```
2. coldnew@Fevia:~/modern-cljs (bash)
coldnew@Fevia ~/modern-cljs $
coldnew@Fevia ~/modern-cljs $
coldnew@Fevia ~/modern-cljs $ boot -V > boot.properties
coldnew@Fevia ~/modern-cljs $
coldnew@Fevia ~/modern-cljs $
coldnew@Fevia ~/modern-cljs $ cat boot.properties
#http://boot-clj.com
#Wed Aug 30 07:57:37 CST 2017
BOOT_CLOJURE_NAME=org.clojure/clojure
BOOT_CLOJURE_VERSION=1.8.0
BOOT_VERSION=2.7.1
coldnew@Fevia ~/modern-cljs $
```

boot.properties 這檔案可以指定該專案使用的 boot 版本，以及執行 boot 使用的 Clojure 版本，避免因為 boot 升級導致專案爆炸

打開你的 target/index.html 看看



Tutorial 2


Immediate Feedback Principle

(second edition)

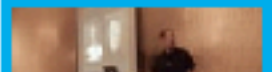
Browser window showing a Vimeo video player. The address bar displays `vimeo.com`. The Vimeo logo and navigation links (Join, Log In, Host videos, Watch, On Demand) are visible. A search bar and an Upload button are also present.

The video player shows a man (Bret Victor) standing on a stage, presenting. A large screen behind him displays two portraits. The video progress bar indicates 54:20.

Bret Victor - Inventing on Principle
5 years ago | More

 **CUSEC** [+ Follow](#)

Recommended
☒ Autoplay next video

 **Bret Victor - Inve...**

[Bret Victor - Inventing on Principle](#) [Bret Victor - Inventing on Principle](#)

Boot tasks

- boot-http
 - 建立 HTTP 伺服器 (像 Python 的 SimpleHttpd)
- boot-reload
 - 提供動態重載靜態檔案功能 (css, js, 圖片)
- boot-cljs-repl
 - 提供 cljs REPL 功能

CLJ-based HTTP server

修改 build.boot

```
(set-env!
  :source-paths #{"src/cljs"}
  :resource-paths #{"html"}

  :dependencies '[[adzerk/boot-cljs "1.7.228-2"]
                  [pandeiro/boot-http "0.7.6"]           ;; add http dependency
                  [org.clojure/tools.nrepl "0.2.12"]]) ;; required by boot-http

(require '[adzerk.boot-cljs :refer [cljs]]
         '[pandeiro.boot-http :refer [serve]]) ;; make serve task visible
```

> NOTE 3: because of a bug in the current `0.7.6` version of `boot-http` we
> had to add the `0.2.12` version of `tools.nrepl` as well.

```
boot serve -d target
2015-10-26 21:38:48.489:INFO:oejs.Server:jetty-7.6.13.v20130916
2015-10-26 21:38:48.549:INFO:oejs.AbstractConnector:Started SelectChannelConnector@0.0.0.0:3000
<< started Jetty on http://localhost:3000 >>
```