

D3.js 是一個利用資料集來操作文件的 JavaScript 函式庫，可以讓你繪製出用 HTML、SVG 及 CSS 做出來的多元化動態資料圖形，D3 遵照著網頁標準，因此現今的瀏覽器皆可直接支援瀏覽，不需要另外安裝任何框架或是套件，D3 很強大的一個功能就是資料視覺化和資料驅動(data-driven) 的 DOM 操作。

資源

- [D3 簡介](#)
- [範例藝廊](#)
- [教學區](#)
- [API 參考文件](#)
- [Release Notes](#)
- [附加元件](#)
- [d3.js on Stack Overflow](#)
- [d3.js Google Group](#)

其它語言 (非官方)

- [English](#)
- [한국어](#)
- [日本語](#)
- [簡體中文](#)
- [Русский](#)
- [Türkçe](#)

瀏覽器支援

D3 能夠在多數現今較先進的瀏覽器(modern browser)上執行，這表示只要你不是用 IE8 或更老舊的瀏覽器就沒問題，D3 開發時的測試平台為 Firefox, Chrome, Safari, Opera, IE9+, Android 和 iOS，有些 D3 在舊一點的瀏覽器可以正常運作，因為 D3 的核心程式庫所需要的最低執行條件就只要你的瀏覽器能夠執行 JavaScript 和 [W3C DOM API](#)。

D3 使用的選擇器是 [Selectors API](#) Level 1, 你也可以預先載入 [Sizzle](#) 來使用，它是相容的，如果你想要使用 SVG(<http://www.w3.org/TR/SVG/>) 或 [CSS3 Transitions](#)，你需要有個較新的瀏覽器，D3 並沒有向下相容。

D3 也可以在 [Node.js](#) 上執行，使用 `npm install d3` 安裝，接著就可以用 `require("d3")` 載入模組，Node 中 DOM 主要由 [JSDOM](#) 來達到，不過只含有限的 DOM 操作，也可以用 [WebWorker](#) 來執行。

安裝

下載最新版本：

- <https://github.com/mbostock/d3/releases>

或是在 HTML 中直接載入最新版本：

```
<script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
```

附註：未壓縮的原始碼包含了 non-ASCII 字元集，必須要以 UTF-8 編碼，可以透過在 `<script>` 的標籤中加入 `charset=utf-8` 屬性，或是加入 `<meta charset="utf-8">`，如果出現了 `SyntaxError: Unexpected token ILLEGAL at var i€ = Math.PI`，這是因為未壓縮的原始碼沒有正確的設定成 ISO-8859-1 編碼，請參閱 [StackOverflow answer](#)。

如果你需要完整的專案(包含測試)，可以下載或複製 D3 的 Git:

- <https://github.com/mbostock/d3/zipball/master>

你也可以透過其它套件管理程式取得 D3，包含 [NPM](#) (Node.js)、[Bower](#)、[Browserify](#)、[Component](#)、[Jam](#)、[Composer / Packagist](#) (PHP)、[SPM](#)、[JSPM](#)、[NuGet](#) (.Net) 和 [AMD](#) (e.g., [RequireJS](#))，官方釋出的 D3 只能用 [NPM](#) 或 [GitHub](#) 取得，其它套件管理程式的支援都是貢獻者做的成果。

使用方式

在開發的時候，你的瀏覽器可能需要強制設定較嚴格的權限，才能夠讀取你檔案系統上的檔案，**如果你在本地端使用 [d3.xhr](#)，你必須要有個本地端的網頁伺服器(web server)**，例如使用 Python 內建的伺服器：

```
python -m SimpleHTTPServer 8888 &
```

或 Python 3+

```
python -m http.server 8888 &
```

如果你已經有安裝 PHP，你可以試試看

```
php -S localhost:8888
```

如果你有執行 Ruby 的環境，可以

```
ruby -run -e httpd . -p 8888
```

當程式跑起來以後，打開瀏覽器連到 <http://localhost:8888/> 就可以看到執行結果了

如果你是用 nodejs

```
npm install http-server -g
http-server
```

其它可行方案是啟動一個本地端的 jetty 實體 (jetty instance)，利用 jetty-runner library，它會在你的系統上的 JVM 上執行，下載 [jetty-runner](#) 後：

```
java -jar jetty-runner-9.3.0.M0.jar --port 8080 .
```

接著就會啟動伺服器，以目前所在的資料夾為根目錄，連結到 <http://localhost:8080>，即可瀏覽執行結果。

D3 支援非同步模組載入的 API (AMD)，例如使用 [RequireJS](#)，你需要像這樣載入：

```
require.config({paths: {d3: "http://d3js.org/d3.v3.min"}});

require(["d3"], function(d3) {
  console.log(d3.version);
});
```

協助開發

如果你想修改 D3 的實作，按下右上方的 "fork" 按鈕來複製一份屬於你的 repository，接著用命令列來 clone 專案到你的電腦中，記得把 *username* 改成你的 GitHub 使用者名稱：

```
git clone git://github.com/username/d3.git
```

如果你要用 D3 來做視覺化應用，建議你要獨立在一個專案中開發，另一方面，假如你想要為 D3 擴充新的功能、修 bug 或執行測試，你應該要 [fork the D3 repository](#)，並安裝 [Node.js](#) (v0.10.x 或更新的版本)，然後在 D3 repository 的根目錄中安裝 D3 的相依套件：

```
npm install
```

執行測試：

```
make test
```