# 大綱

在chrome extension 中的html 文件，大概有三種

* browser page(在某個tab 下的網頁，例如第2個圖中的紅框)
* content script(注射到當前tab 的程式碼)
* background page(可能是js,或html)
* UI page 當browser action 上的某個按鈕被按下時，彈出的社窗(例如圖中的popup.html)

([參考](https://developer.chrome.com/extensions/overview#arch))

圖1



圖2



圖3

# 簡單範例

## 範例：最少程式碼的 chrom extension

Bare minimum Chrome extension to inject a JS file into the given page when you click on the browser action icon. The script then inserts a new div into the DOM.

##### 【background.js】

當extension 上的按鈕被按的時候，看看是哪一個文件(tab)要被注射'inject.js'。

// this is the background code...

// listen for our browerAction to be clicked

chrome.browserAction.onClicked.addListener(**function** (tab) {

 // for the current tab, inject the "inject.js" file & execute it

 chrome.tabs.executeScript(tab.ib, {

 file: 'inject.js'

 });

});

##### 【inject.js 】

// this is the code which will be injected into a given page...

(**function**() {

 // just place a div at top right

 **var** div **=** document.createElement('div');

 div.style.position **=** 'fixed';

 div.style.top **=** 0;

 div.style.right **=** 0;

 div.textContent **=** 'Injected!';

 document.body.appendChild(div);

 alert('inserted self... giggity');

})();

##### 【 manifest.json】

{

 "name": "Injecta",

 "version": "0.0.1",

 "manifest\_version": 2,

 "description": "Injecting stuff",

 "homepage\_url": "http://danharper.me",

 "background": {

 "scripts": [

 "background.js"

 ],

 "persistent": **true**

 },

 "browser\_action": {

 "default\_title": "Inject!"

 },

 "permissions": [

 "https://\*/\*",

 "http://\*/\*",

 "tabs"

 ]

}

## 範例2

在每篇網頁上，把某個關鍵字打光。

##### 【 manifest.json】

{

 "name": "Highlight some phrases",

 "description": "Hightlight some pre defined text from websql database after page loads",

 "version": "0.1",

 "permissions": [

 "tabs","<all\_urls>"

 ],

 "browser\_action": {

 "default\_icon": "icon.png"

 },

 "content\_scripts": [

 {

 "matches": [

 "http://\*/\*",

 "https://\*/\*"

 ],

 "js": ["content.js"],

 "run\_at": "document\_end" // pay attention to this line

 }

 ],

 "manifest\_version":2

}

##### 【content.js】

str**=**"regexp"

//var regex=/hello/g;

**var** regex **=** **new** RegExp(str, "gi");

**var** estr**=**document.body.outerHTML;

**var** result**=**estr.replace(regex,"<font color='red'>"**+**str**+**"</font>")

document.body.outerHTML**=**result;

## Manifest

每一個 Chrome Extension 都會有一個 JSON 格式的設定檔(manifest.json)，裡面會定義這個 Extension 的名稱、版本、類型、權限等等。

### 必要的設定

manifest 裡至少要設定這三項，如下。

manifest.json

{

"manifest\_version": 2,

"name": "Extension 名稱",

"version": "Extension 版本"

}

解析manifest\_version的版本，現在只允許 2 的版本。

### 官方建議設定

官方建議設定三項default\_locale, description, icons，如下

manifest.json

{

"default\_locale": "en",

"description": "專案描述",

"icons": {

"16": "icon16.png",

"48": "icon48.png",

"128": "icon128.png"

},

}

* 如果沒有包裝成多國語言打算的話，不用 default\_locale 這個欄位，如果有設定，就必須有 「專案\_目錄/\_locales/en/messages.json」。
* description 不能超過 132 個字元。
* icons 有三種大小，分別是 16x16, 48x48, 128x128，可以不設定，此時是預設圖。

### 顯示類型的設定

Extension 有三種類型(只能設定其中一種)，

* browser\_action ：顯示在瀏覽器右上角
* page\_action ：顯示在網址列。
* 不顯示：前兩項都不設定。

Note:

* 前兩者都可設定ICON、標題、彈窗。
* 如果extension專案只限定在某些網頁執行（利用match 欄位）就選擇 page\_action，否則選擇 browser\_action。

manifest.json

{

"browser\_action": {...},

"page\_action": {...},

}