JavaScript – Creating A Downloadable Link in the Browser

# 大綱

怎樣建立 dataurl

參考範例2.

# Creating A Downloadable File in the Browser

參考

<[本文](http://cwestblog.com/2014/10/21/javascript-creating-a-downloadable-file-in-the-browser/)>

**目的是自動產生一個如下的連結：**

只是利用data URIs.例如，clicking here

。 產生內容為 Hello World!!!. 的連結，並提供下載連結：

 <a href= "data:text/plain;charset=utf-8,Hello%20world!!!">下載</a>。

檔案名稱new 2.html

**<html>**

 **<body>**

 **<a** href**=** "data:text/plain;charset=utf-8,Hello%20world"**>**下載**</a>**

 **</body>**

**</html>**

會出現：



|  |
| --- |
| Chrome 禁止直接從頁面打開 Data URI 網址了，只能利用右鍵傳統的釣魚網站通常利用相似的URL來冒充真正的網站，比如 yahao.com。釣魚網址一般是假冒網站，來獲取使用者資料。雖然browser會利用黑名單（比如google的 Safe Browsing）提示不安全。但是，利用Data URI 可以跳過這些檢測？可以參考網路安全公司公布的利用 Data URI 來釣魚的真實案例（2016年，將近85%的資金損失是通過釣魚網址洩露支付資訊造成的）。利用 Data URI，把整個網頁都放到這個 URL。所以黑名單遮罩技術對他無效。 |

**只是如何讓他自動產生?**

【範例1】：

<!DOCTYPE html**>**

**<html>**

**<head>**

 **<meta** charset**=**"utf-8"**>**

 **<title>**JS Bin**</title>**

 **<script>**

**function** main() {

 setupDownloadLink **=** **function**(link, code) {

 link.href **=** 'data:text/plain;charset=utf-8,' **+** encodeURIComponent(code);

 };

 window.onload **=** **function**() {

 txt.value **=** main **+** ''; // txt.value <1>

 }

}

main();

 **</script>**

**</head>**

**<body>**

 **<div** id**=**"txtWrap"**>**

 **<textarea** id**=**"txt"**></textarea>**

 **</div>**

 **<a** href**=**"" id**=**"link" onclick**=**"setupDownloadLink(this, txt.value)" download**=**"code.txt"**>**Download Above Code**</a>**

**</body>**

**</html>**

註解：

1

* txt 是 textarea 標籤的 [id](#refAnker_984609)。
* window.onload ：文件載入完畢以後觸發。
* 整個的意思是說，載入文件以後在<textaread id="txt"> 中，顯示main函數的內容。

【範例2】：按下Download above code 按鈕以後，將文字框裡面的文字下載為檔案。



<!DOCTYPE html**>**

**<html>**

  **<head>**

    **<style** type**=**"text/css"**>**...**</style>**

    **<script** type**=**"text/JavaScript"**>**

    window.onload **=** **function**() {

      **var** txt **=** document.getElementById('txt');

      txt.value **=** window.onload **+** '';

      document.getElementById('link').onclick **=** **function**(code) {

        **this**.href **=** 'data:text/plain;charset=utf-8,'

          **+** encodeURIComponent(txt.value);

      };

    };

    **</script>**

  **</head>**

  **<body>**

    **<div** id**=**"txtWrap"**>**

      **<textarea** id**=**"txt"**></textarea>**

    **</div>**

    **<a** href**=**"" id**=**"link" download**=**"code.txt"**>**Download Above Code**</a>**

  **</body>**

</html>

提要：整個HTML的重點，只是要在href 中填入：

"'data:text/plain;charset=utf-8," + <我們要輸出的文字>

執行結果



除了encodeURIComponent() ，也可以使用 escape() ，只是這個函數好像不是標準？

|  |
| --- |
| encodeURIComponent:將字串編碼為utf-8 格式。URL 或是 URI 通常是一個資源的位址，由可讀字串組成，例如，<http://www.nkfust.edu.tw>但是，有些情況例如，[http://www.nkfust.edu.tw/foo?hello="xxxxxx](http://www.nkfust.edu.tw/foo?hello=%22xxxxxx)" 中的 xxxx可能是某些具有特殊意義的字串，此時就必須編碼。另外，內嵌URL，例如data:text/plain;charset=utf-8,Hello%20world上例中，本來的字串應該是 Hello world 經過encodeURIComponent() 編碼以後會將空白轉為%20 。例如，var world = "A string with symbols & characters that have special meaning?";var uri = 'http://example.com/foo?hello=' + encodeURIComponent(world);相對於encodeURIComponent 還有一個函數叫做encodeURI:encodeURI 的參數，預計是一個完整的URI 但是某些字元需要編碼。 |

討論：上面的例子中,code 沒用到