

jQuery

王維駿

簡介

- 開源的JavaScript函式庫
- 跨瀏覽器
- 功能豐富
- 語法簡潔
- 擴充性高

jQuery 與 JavaScript

- jQuery的誕生，是為了讓你更方便寫JavaScript
 - 不是取代，本質上一樣是JavaScript
 - 相同的事情，更加簡便的寫法

核心概念

- 查詢 – 透過「選擇器」
 - 類似CSS選擇器，但是功能又多了一些
 - 每一次查詢都會回傳 jQuery 集合物件，包含n個Dom節點
- 操作 – 透過 jQuery 提供的函式
 - 操作針對整個集合物件內的所有Dom節點
 - 函式有內建(Built-in)或插件(Plug-in)

掛載OnClick事件

- JavaScript

```
document.getElementById("go").onclick =  
function () {  
    // do something  
};
```

- jQuery

```
$("#go").click(function () {  
    // do something  
});
```

創立元素操作比較

- JavaScript

```
var listItem = document.createElement("li");  
listItem.className = "list-group-item";  
listItem.innerHTML = "<div><a>...</a></div>";  
document.getElementById("results").appendChild(listItem);
```

- jQuery

```
$("<li></li>").addClass("list-group-item")  
    .html("<div><a>...</a></div>").appendTo("#results");
```

取值操作比較

- JavaScript

```
for (let i = 0; i < elements.length; i++) {  
    if (elements[i].type === 'input' &&  
        elements[i].className === 'name') {  
        value = elements[i].value;  
        break;  
    }  
}
```

- jQuery

```
$('input.name').val()
```

Input 欄位操作比較

- JavaScript

```
var input = document.getElementById("input");  
input.value = 0;
```

- jQuery

```
var $input = $("#input");  
$input.val(0);
```


innerHTML 操作比較

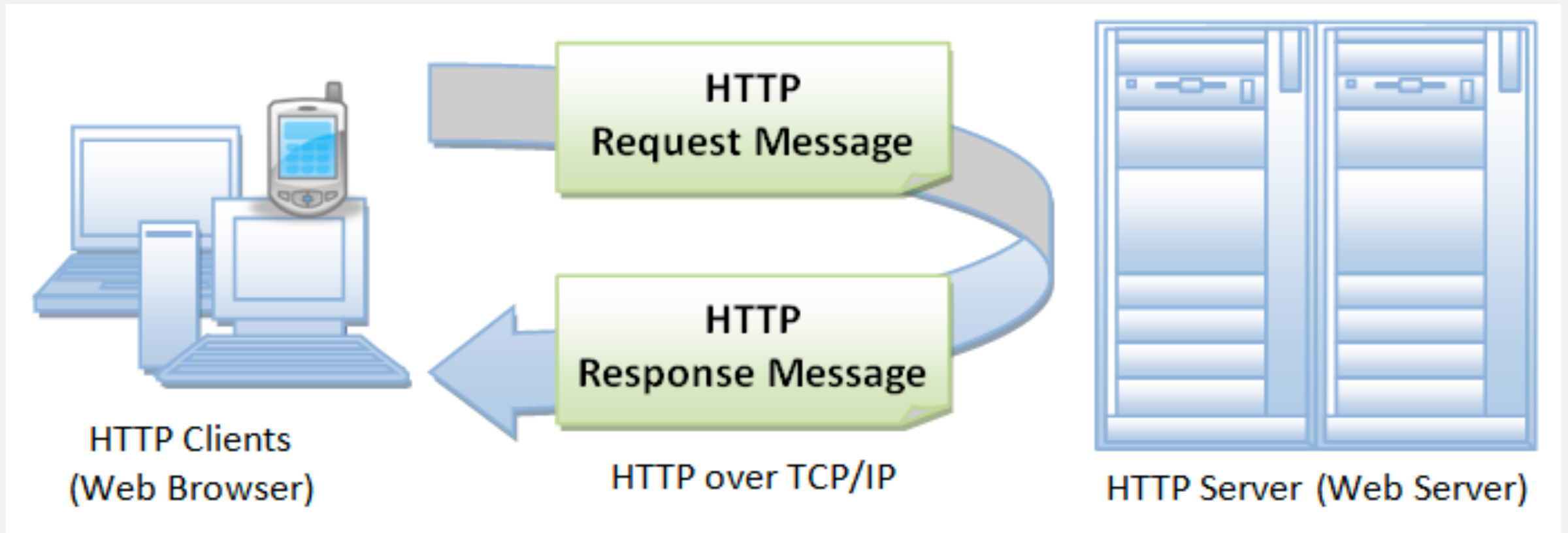
- JavaScript

```
var Results = document.getElementById("Results");  
Results.innerHTML = "";
```

- jQuery

```
var $Results = $("#Results");  
$Results.html("");
```

HTTP



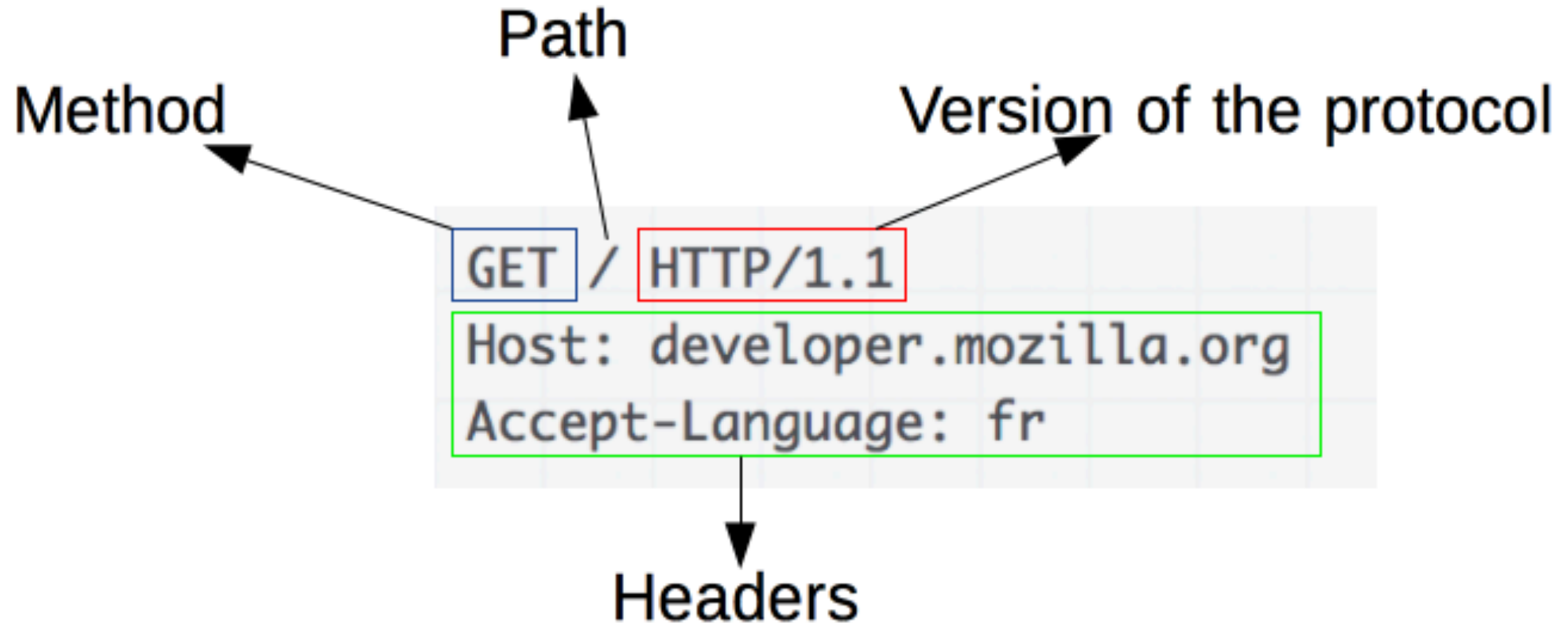
HTTP 是什麼？

- HTTP是一個用戶端和伺服器端之間請求和應答的標準
- 全球資訊網（Web）的基礎協定
 - 規範 Web 的資料通訊格式
 - 基於 Socket（二進位網路） / TCP 上的一個通訊規範

HyperText Transfer Protocol

- HTTP/2
 - 最新版本，2015年5月作為網際網路標準正式發布
- HTTP/1.1
 - 目前主流(1997)
- HTTP/1.0
 - 至今仍被廣泛採用，特別是在代理伺服器中
- HTTP/0.9
 - 已過時，只支援Get方法

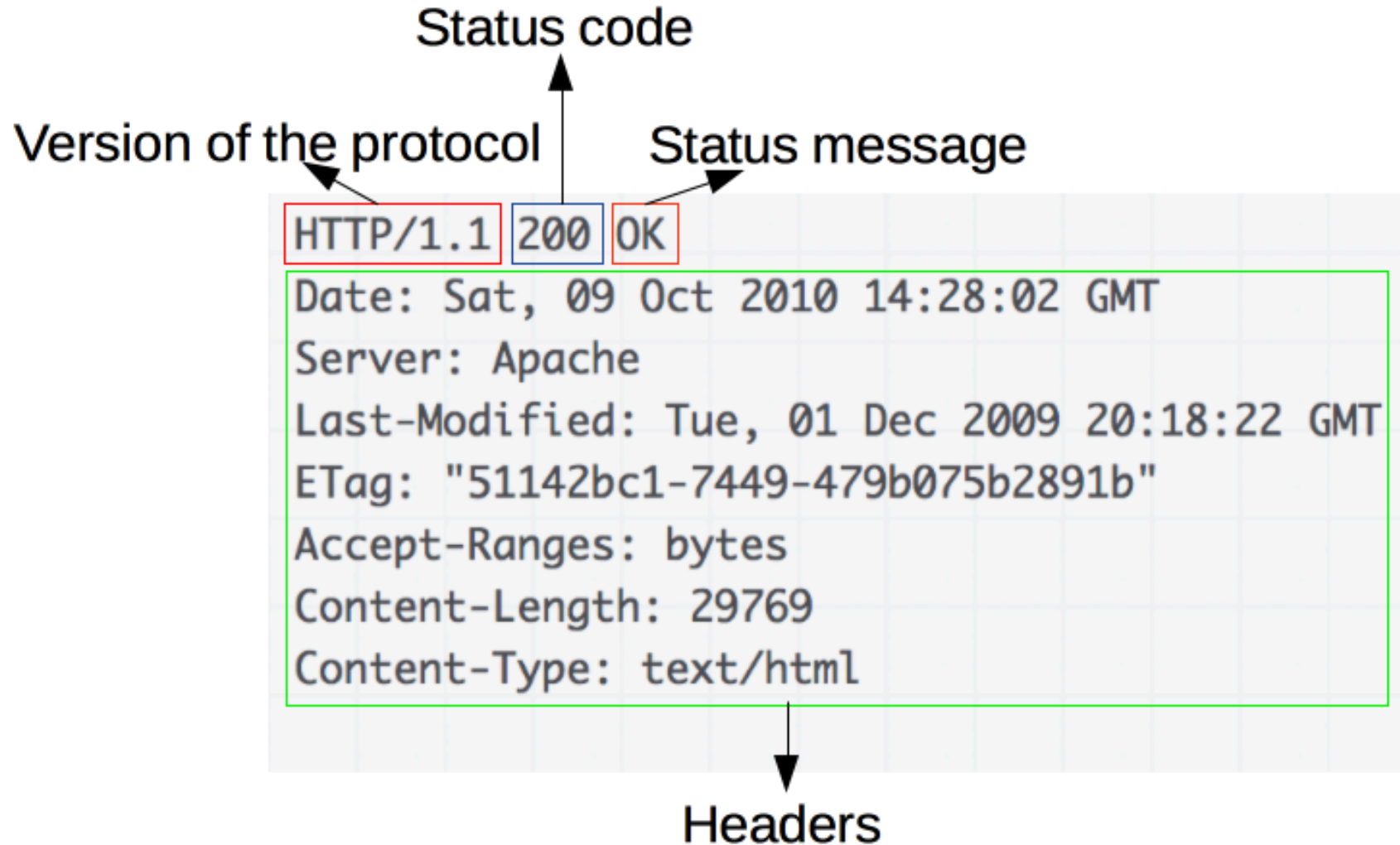
HTTP Request



HTTP Request

- 一個HTTP的method，經常是由一個動詞像GET、POST來定義客戶端的動作行為。
- 獲取資源的路徑，通常是元素資源的URL
- HTTP協議版本號
- 為服務端表達其他信息的可選頭headers。
- 像POST這樣的方法，body就包含了發送的資源

HTTP Response



HTTP Response

- HTTP協議版本號。
- 狀態碼（ `status code` ），對應請求執行成功或失敗，
- 狀態信息是非權威的狀態碼描述信息，可自行設定。
- HTTP headers，與請求頭部類似。
- 比起Request，Response中常包含獲取的資源body。

XML HTTP Request (XHR)

- 最早賦予 JavaScript 網路功能的 API
- 允許由 JavaScript 發出 HTTP 要求並接取回應

XHR

```
var xhr = new XMLHttpRequest(); // 初始化 XHR
xhr.open("GET", "WebAPI"); // 開啟連線
xhr.onload = function () { // 回應事件處理
    if (200 <= xhr.status && xhr.status <= 299) {
        // 取得回應的 Content-Type 表頭欄位
        // 以決定如何處理回應
        var type = xhr.getResponseHeader("Content-Type");
        // 建構元 (strategy, response)
        var handler;
        // 使用簡易的 正規表達式，判斷媒體類型
        if (type.match(/^application\/json/)) {
            handler = new Handler(jsonHandler, JSON.parse(xhr.responseText));
        } else if (type.match(/^application\/xml/)) {
            handler = new Handler(textHandler, xhr.responseXML);
        } else {
            handler = new Handler(textHandler, xhr.responseText);
        }
        handler.handleResponse();
    }
}
xhr.send(); // 提交要求給伺服器
```

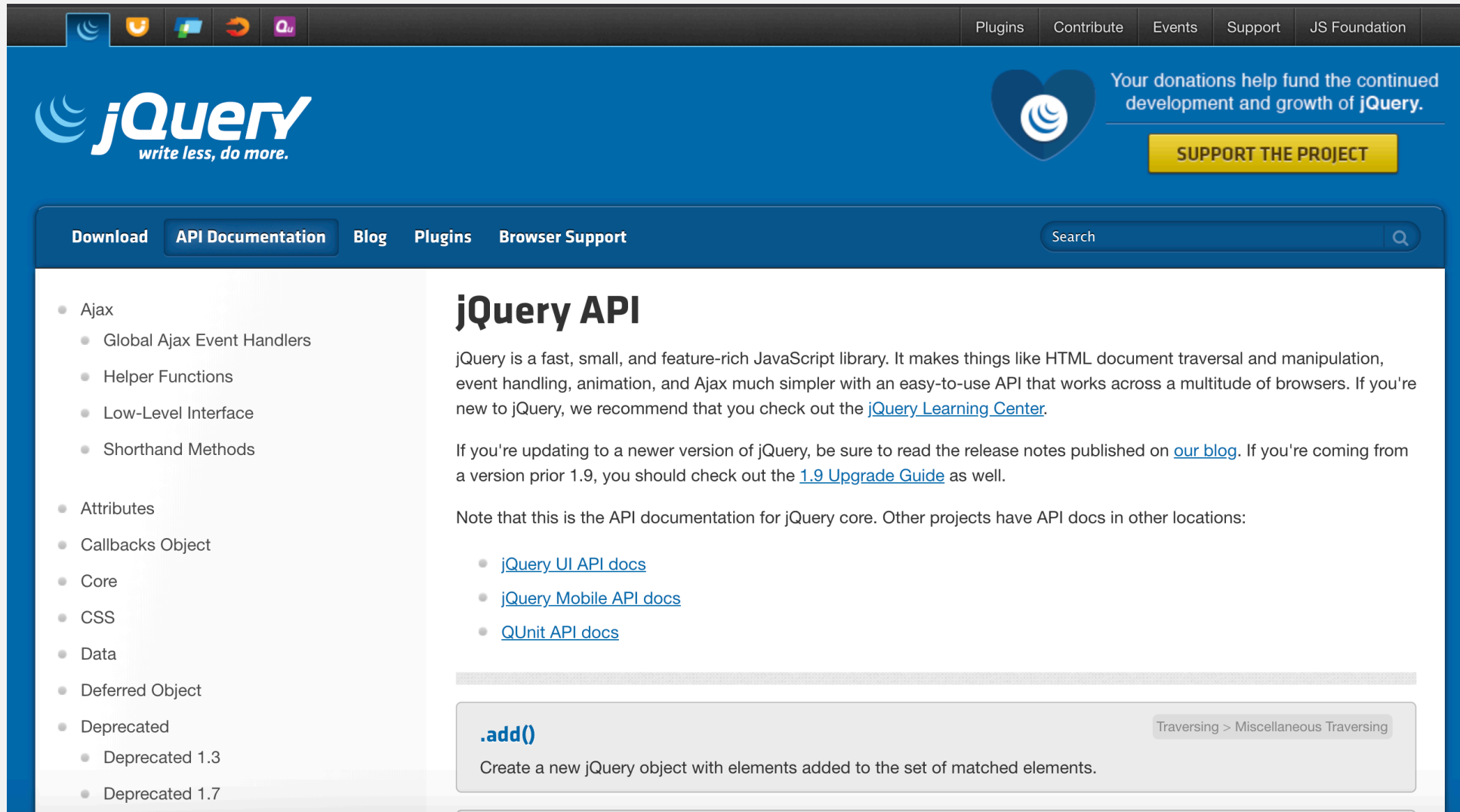
Asynchronous JavaScript and XML (AJAX)

- 運用XHTML+CSS來表達資訊
- 運用JavaScript操作DOM執行動態效果
- 運用XML和XSLT操作資料
- 運用XMLHttpRequest或新的Fetch API與網頁伺服器進行非同步資料交換

Ajax

```
$.ajax({  
    type: "method", //Get、Post 等等  
    url: "url", //WebAPI位置  
    data: "data", //要傳遞的參數  
    dataType: "dataType", //要回傳的格式  
    success: function (response) {}, //成功  
    error: function(response){}, //失敗  
});
```

查詢jQuery文件 (難用)



The screenshot shows the jQuery website's API documentation page. At the top, there's a navigation bar with links for Plugins, Contribute, Events, Support, and JS Foundation. Below this is the jQuery logo with the tagline "write less, do more." and a heart icon with the text "Your donations help fund the continued development and growth of jQuery." and a "SUPPORT THE PROJECT" button. The main navigation bar includes links for Download, API Documentation (which is highlighted), Blog, Plugins, and Browser Support. A search bar is also present. On the left side, there's a sidebar with a list of categories: Ajax, Attributes, Callbacks Object, Core, CSS, Data, Deferred Object, Deprecated, and Shorthand Methods. The main content area is titled "jQuery API" and contains an introduction to the library, a link to the jQuery Learning Center, and information about updating to newer versions. It also lists other API documentation locations: jQuery UI API docs, jQuery Mobile API docs, and QUnit API docs. At the bottom, there's a section for the ".add()" method, which is described as "Create a new jQuery object with elements added to the set of matched elements."

jQuery API

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. If you're new to jQuery, we recommend that you check out the [jQuery Learning Center](#).

If you're updating to a newer version of jQuery, be sure to read the release notes published on [our blog](#). If you're coming from a version prior 1.9, you should check out the [1.9 Upgrade Guide](#) as well.

Note that this is the API documentation for jQuery core. Other projects have API docs in other locations:

- [jQuery UI API docs](#)
- [jQuery Mobile API docs](#)
- [QUnit API docs](#)

.add() Traversing > Miscellaneous Traversing

Create a new jQuery object with elements added to the set of matched elements.

查詢jQuery文件 (好用)

jQuery Quick API Reference					3.0	Search...	Preferences...
SELECTORS			ATTRIBUTES / CSS		MANIPULATION	TRAVERSING	EVENTS
Basics * .class element #id selector1, selectorN, ...	Visibility Filters :hidden :visible Attribute [name ="value"] [name*="value"] [name~="value"] [name\$="value"] [name="value"] [name!="value"] [name^="value"] [name="value"][name2="value2"]	Forms :button :checkbox :checked :disabled :enabled :focus :file :image :input :password :radio :reset :selected :submit :text	Attributes .attr() .prop() .removeAttr() .removeProp() .val() CSS .addClass() .css() jQuery.cssHooks jQuery.cssNumber jQuery.escapeSelector() .hasClass() .removeClass() .toggleClass()	Dimensions .height() .innerHeight() .innerWidth() .outerHeight() .outerWidth() .width() Offset .offset() .offsetParent() .position() .scrollLeft() .scrollTop() Data jQuery.data() .data() jQuery.hasData() jQuery.removeData() .removeData()	Copying .clone() DOM Insertion, Around .wrap() .wrapAll() .wrapInner() DOM Insertion, Inside .append() .appendTo() .html() .prepend() .prependTo() .text() DOM Insertion, Outside .after() .before() .insertAfter() .insertBefore() DOM Removal .detach() .empty() .remove() .unwrap() DOM Replacement .replaceAll() .replaceWith()	Filtering .eq() .filter() .first() .has() .is() .last() .map() .not() .slice() Miscellaneous Traversing .add() .addBack() .andSelf() .contents() .each() .end() Tree Traversal .children() .closest() .find() .next() .nextAll() .nextUntil() .parent() .parents() .parentsUntil() .prev() .prevAll() .prevUntil() .siblings()	Browser Events .effect() .resize() .scroll() Document Loading .load() .ready() .unload() Event Handler Attachment .bind() .delegate() .die() .live() .off() .on() .one() .trigger() .triggerHandler() .unbind() .undelegate() Form Events .blur() .change() .focus() .focusin() .focusout() .select() .submit() Keyboard Events .keydown() .keypress() .keyup()