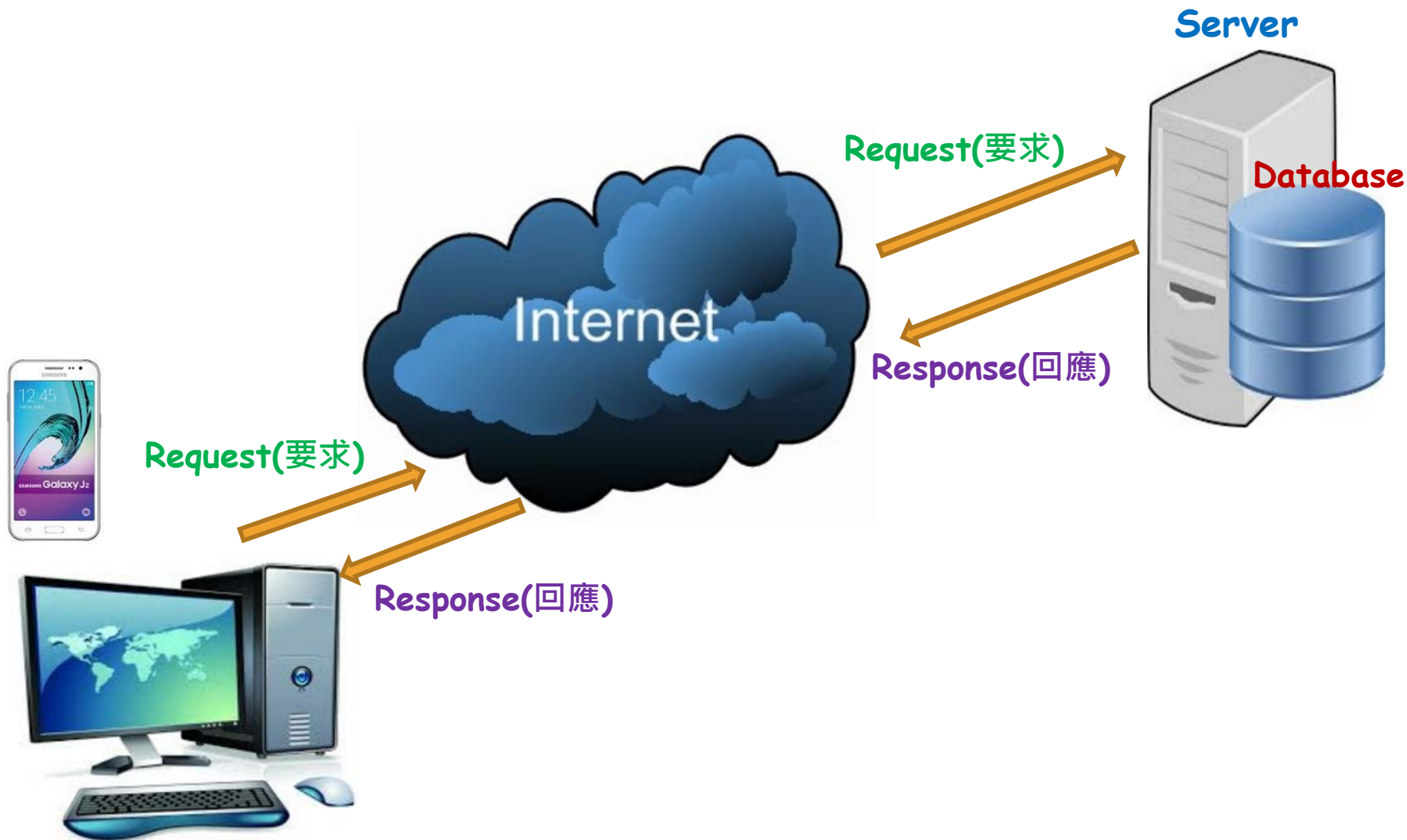


# 伺服器網頁程式

---

William Wang

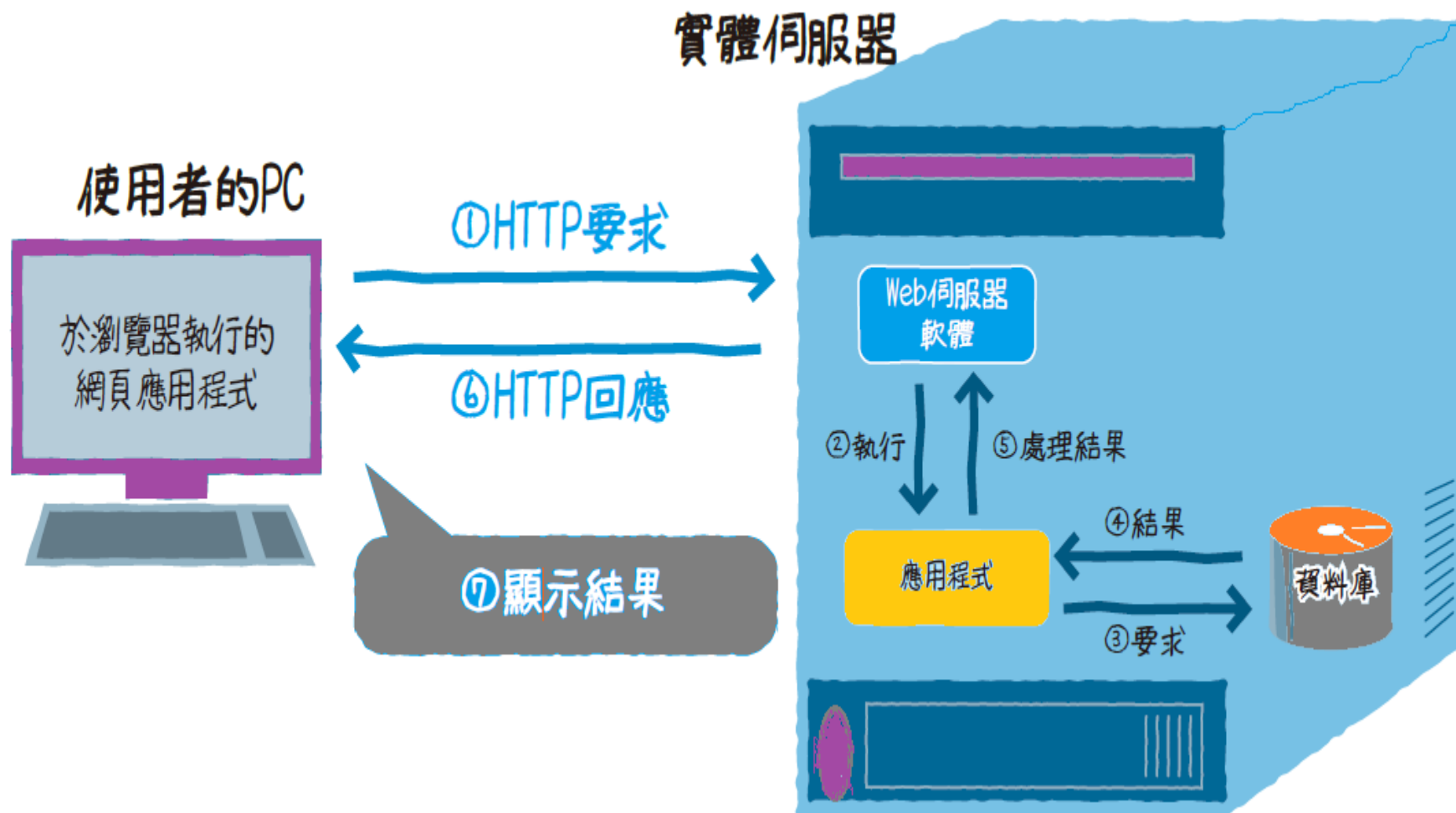
# 網頁應用程式架構

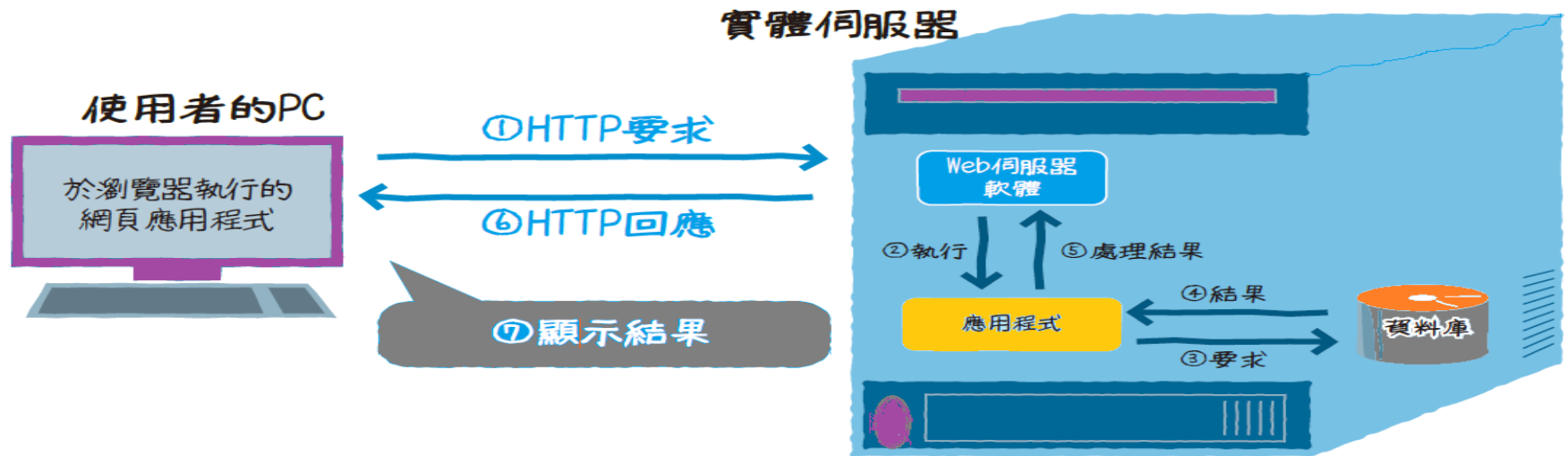


# 網頁應用程式的程式

- 網頁應用程式的程式可分成兩種，第一種是於網頁伺服器執行的「伺服器端應用程式」。具代表性的伺服器端應用程式語言有**PHP**、**Python**、**Ruby**、**Perl**。伺服器端程式在接收到使用者（瀏覽器）的要求後，會根據該要求進行處理，再將結果傳給使用者。
- 另一個是於瀏覽器執行的「用戶端應用程式」。用戶端應用程式可與建構文件的「**HTML**」以及指定**HTML**樣式的「**CSS**」合併使用。近年來，用戶端應用程式的設計也以**JavaScript** 為主流。

# 網頁應用程式的處理流程





- ① Browser向Server發出要求（HTTP 要求）。
- ② 網頁伺服器判斷來自瀏覽器的要求再執行應用程式。
- ③ 接收要求的應用程式將根據要求的內容將要求傳遞給資料庫。
- ④ 接收要求的資料庫將儲存與取得資料，再將結果傳回應用程式。
- ⑤ 應用程式內的處理與根據資料庫結果進行處理,結果將傳回web server。
- ⑥ 網頁伺服器將應用程式的結果傳給Browser（HTTP 回應）。
- ⑦ 於Browser顯示結果。

# 所使用的軟體

- Apache, php, MySQL (or Maria), phpMyadmin  
p.s. The above software can be found in XAMPP.

# 所使用的程式

- html: 前端 user
- css: 前端 user
- JavaScript: 前端 user 或 後端操作 資料庫
- php: 後端操作 資料庫

# php Sources

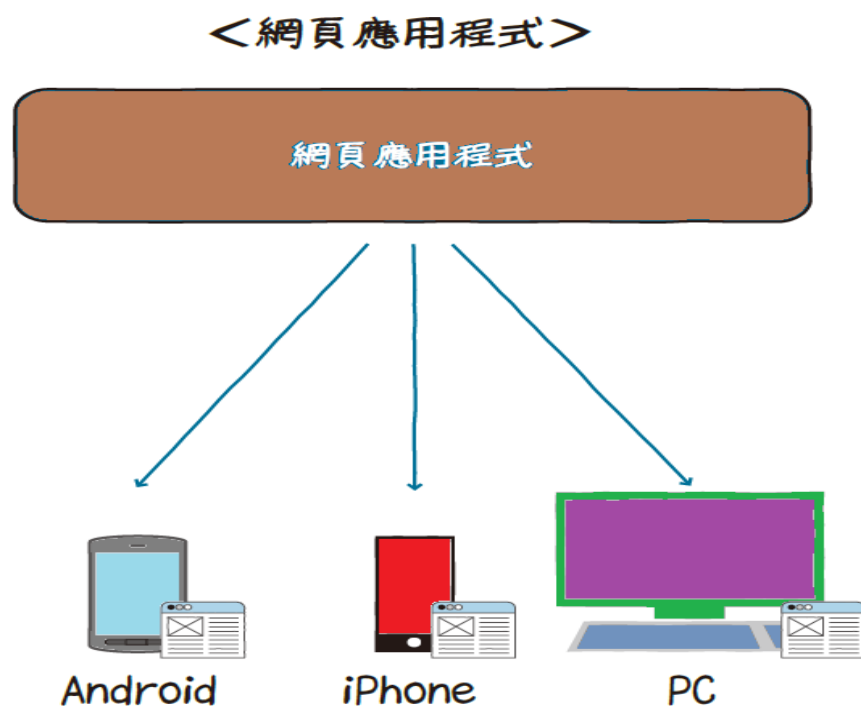
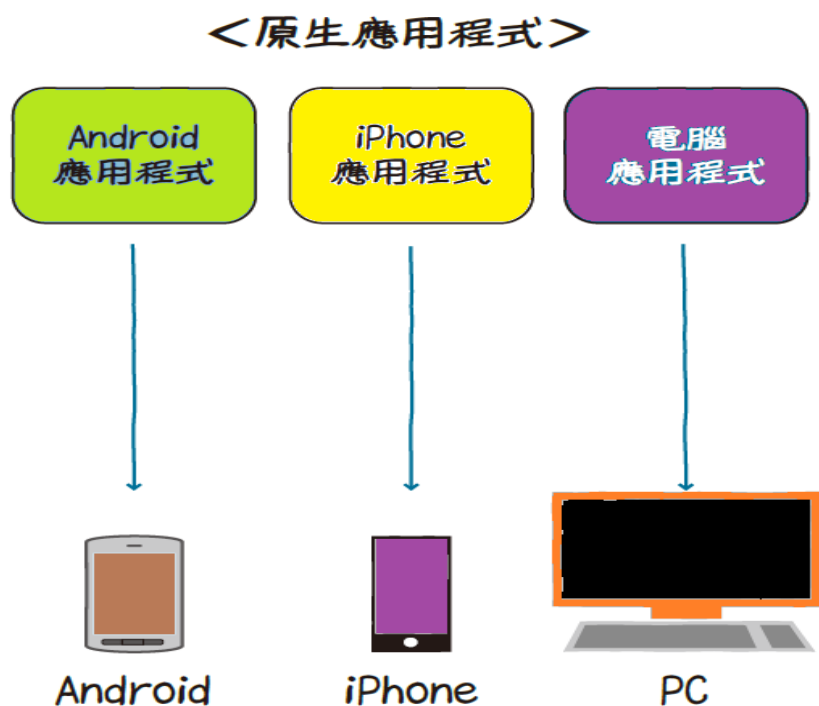
- <https://www.w3schools.com/php/>
- [https://www.tutorialspoint.com/php7/php7\\_csprng.htm](https://www.tutorialspoint.com/php7/php7_csprng.htm)



# 網頁應用與原生應用程式

# 程式上的差異

- 原生型應用程式（Native App）主要意指，利用該平台官方建議的程式語言來開發應用程式，舉例來說：iOS使用的是Objective-C，而Android的是Java，所以針對不同的平台，需要不同的開發方式。
- 網頁應用程式，則依附且在Browser上執行



# 發佈方式的不同

- 就上述而言, 原生應用程式就是透過AppStore 或Google Play 發佈。需要進行審查，所以耗費的時間較久
- 網頁應用程式則可將要發佈的檔案上傳至web server，再提供給使用者使用，開發人員也比較容易控制發佈流程。

# 執行環境與執行速度上之差異

- 原生應用程式不一定需要連上網路，而**網頁應用程式**則必須與網頁伺服器連線。
- 隨著瀏覽器的性能提昇，**網頁應用程式**的執行速度與原生應用程式的差距也慢慢縮小，但是若要顯示大量的多媒體動畫，更新畫面的速度就會遇到瓶頸，感覺執行的速度變慢。

# 相關網頁應用與趨勢

# 有效的前台開發

## <傳統的網頁應用程式>

以伺服器端處理為主軸

②利用ImageMagic進行  
圖片處理或是於資料庫  
新增資料

①直接上傳圖片

③以HTML格式  
將新增的資料  
傳給使用者



## <現在的網頁應用程式>

前台與伺服器端  
分工合作

②新增至資料庫

①利用瀏覽器完成  
圖片處理的階段，  
再以base64格式  
傳送

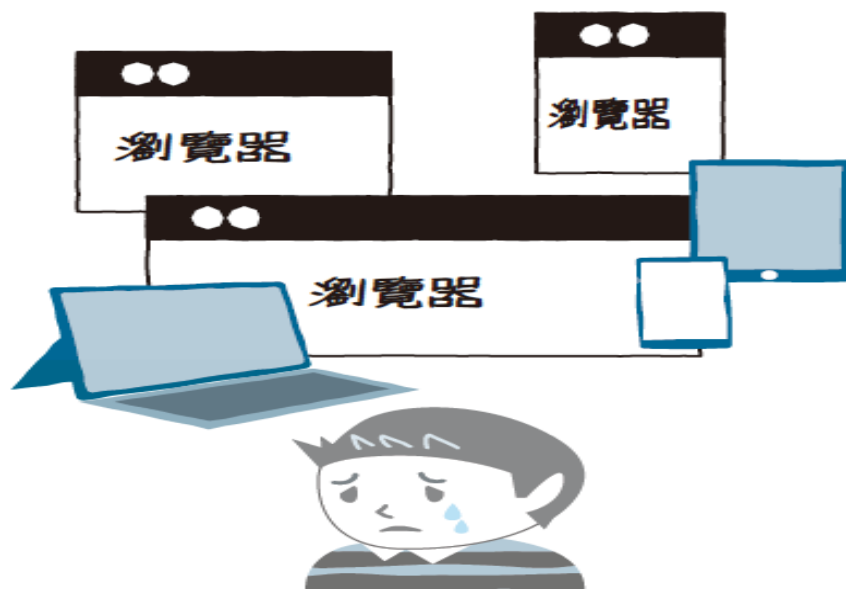
③以JSON格式  
將新增的資料  
傳給使用者

④剖析JSON再以  
HTML的格式顯示



# 以功能為單位來處理

## <以瀏覽器為單位的分歧處理>



瀏覽器的種類何其多，有幾種瀏覽器就必須設立幾個條件式。一旦出現新的瀏覽器，就必須重新改寫應用程式。

## <以功能為單位的分歧處理>



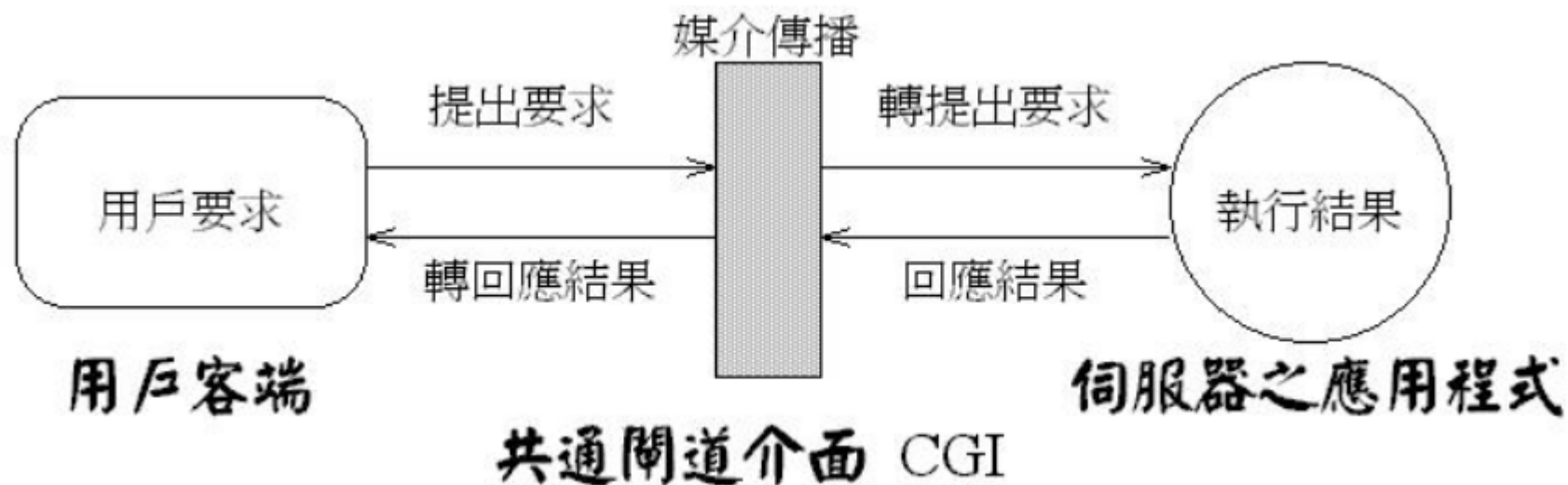
以確認瀏覽器是否具有該功能的方式進行分歧處理。即便新的瀏覽器問世，也因為已經利用功能判斷要進行的處理，就不需要再重寫程式了。

**CGI**



- 原始的HTML語言是設計用來展現**靜態**的資料，它讓人使用一種簡單的語法展現出豐富的多媒體資料，就像廣告看板一樣。由於WWW具有相當大的商業用途，因此推出後大受歡迎。
- 可是單純的靜態展示好像缺少了什麼？如果使用者需要的資料具有**時效性**，必需時常更新，這時該怎麼辦呢？又或者你想**留下使用者的資料**，讓你的網頁能跟**使用者達到互動的效果**，這時又該如何？由於上述種種需求，於是就誕生了CGI。
- CGI是Common Gateway Interface的縮寫，中文翻做『共通閘道介面』。它是一種標準介面程式，能讓你的**網頁**跟**http server**溝通，達到跟使用者互動的效果。而且透過CGI程式，**可以讓你動態的產生網頁，秀出server上的最新資料**。當你**link**到一個**CGI**物件時，你取回的文件並非是一份靜態資料，而是一個藉由程式動態產生的**HTML資料流**。

- **Server** 傳回來的資料也許分分秒秒都在更改，或者針對使用者的特殊查詢而有不同的反應，例如股票市場行情等等。簡單來說，**CGI**程式就是能夠動態產生網頁，並讓一般使用者經由**http** 取用現存在傳統資訊系統內的資料。
- **CGI**程式可以根據使用者輸入的要求自動產生**HTML**格式的資料。因為輸入資料是由**http server**負責接收，故得找出一個有效率的方法，來做使用者輸入參數和產生**HTML**文件程式間的溝通工作。



# 什麼是FastCGI

- FastCGI是在CGI標準協議上發展出來的一個變種協議，它的主要目標是減輕web伺服器與CGI程式之間互動時的負載，這樣一臺伺服器就可以在同一時間處理更多的web請求。

# 什麼是 PHP-FPM

- PHP-FPM (FastCGI Process Manager)是FastCGI在PHP上的具體實現，從PHP5.3.3開始，已經被整合到PHP的安裝包中。

# Apache與php的連線

- CGI ( 基本已經不用 )
- 使用模組化 **FastCGI**

The End