Hw6\_http

HTTP/1.0

為了提高系統的效率，HTTP 1.0規定瀏覽器與伺服器只保持短暫的連線，瀏覽器的每次請求都需要與伺服器建立一個TCP連線，伺服器完成請求處理後立即斷開TCP連線，伺服器不跟蹤每個客戶也不記錄過去的請求。

HTTP 1.1

支援持久連線，在一個TCP連線上可以傳送多個HTTP請求和響應，減少了建立和關閉連線的消耗和延遲。一個包含有許多影像的網頁檔案的多個請求和應答可以在一個連線中傳輸，但每個單獨的網頁檔案的請求和應答仍然需要使用各自的連線。

HTTP 1.1還允許客戶端不用等待上一次請求結果返回，就可以發出下一次請求，但伺服器端必須按照接收到客戶端請求的先後順序依次回送響應結果，以保證客戶端能夠區分出每次請求的響應內容，這樣也顯著地減少了整個下載過程所需要的時間。

HTTP2.0

**多路複用 (Multiplexing)**: 多路複用允許同時通過單一的 HTTP/2 連線發起多重的請求-響應訊息。在 HTTP/1.1 協議中瀏覽器客戶端在同一時間，針對同一域名下的請求有一定數量限制。超過限制數目的請求會被阻塞。因此 HTTP/2 可以很容易的去實現多流並行而不用依賴建立多個 TCP 連線，HTTP/2 把 HTTP 協議通訊的基本單位縮小為一個一個的幀，這些幀對應著邏輯流中的訊息。並行地在同一個 TCP 連線上雙向交換訊息。

**服務端推送（Server Push）**:服務端推送是一種在客戶端請求之前傳送資料的機制。在 HTTP/2 中，伺服器可以對客戶端的一個請求傳送多個響應。Server Push 讓 HTTP1.x 時代使用內嵌資源的優化手段變得沒有意義；如果一個請求是由你的主頁發起的，伺服器很可能會響應主頁內容、logo 以及樣式表，因為它知道客戶端會用到這些東西。這相當於在一個 HTML 文件內集合了所有的資源，不過與之相比，伺服器推送還有一個很大的優勢：可以快取！也讓在遵循同源的情況下，不同頁面之間可以共享快取資源成為可能。

**HPACK 演算法:**HPACK演算法是新引入HTTP/2的一個演算法，用於對HTTP頭部做壓縮。其原理在於：

* 客戶端與伺服器端根據[RFC 7541](https://tools.ietf.org/html/rfc7541)的附錄A，維護一份共同的靜態字典（Static Table），其中包含了常見頭部名及常見頭部名稱與值的組合的代碼。
* 客戶端和伺服器端根據先入先出的原則，維護一份可動態添加內容的共同動態字典（Dynamic Table）。
* 客戶端和伺服器端根據[RFC 7541](https://tools.ietf.org/html/rfc7541)的附錄B，支援基於該靜態哈夫曼碼錶的哈夫曼編碼（Huffman Coding）。