網路程式設計
簡介

Instructor: 馬豪尚

Teams

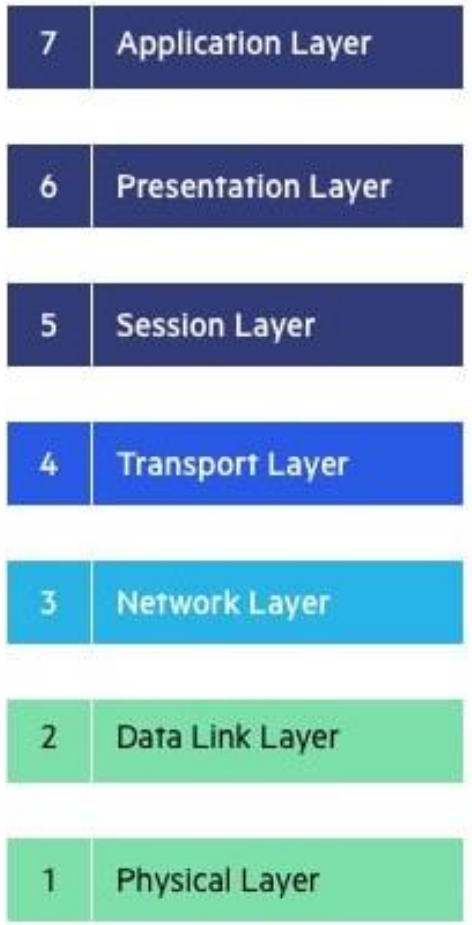
- 團隊代碼
  - **e28aqnd**

什麼是網際網路(Internet)？

- 網際網路實際上並不是真正的網路，它是一個虛擬的概念，是由各種不同網路之間所串而連成的一個單一巨大國際網路，並在其上面提供網路服務。
- 為了能夠將各種不同網路連接起來，這些網路就必須以一組通用的協定相連。

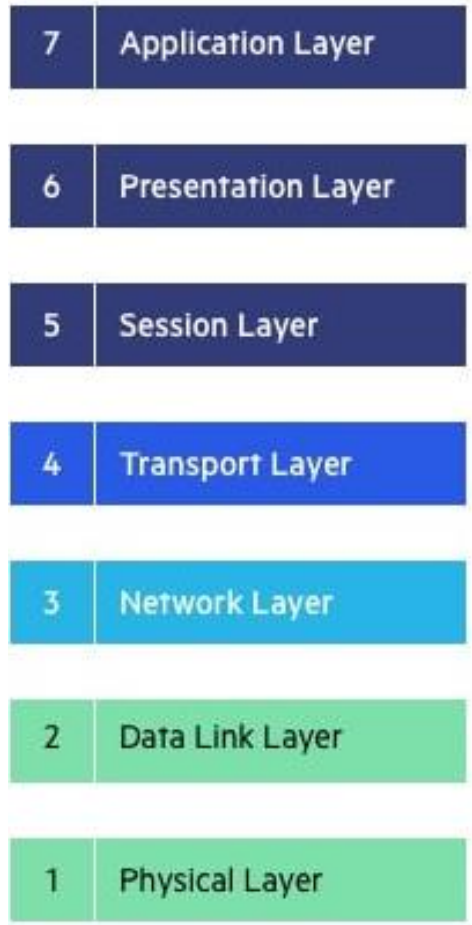
OSI 網路模型7層架構

- 應用層
  - 這層定義使用者的應用程式交換資料的方式
  - Web 瀏覽器和電子郵件客戶端
  - HTTPS、POP 、FTP
- 表示層
  - 這層負責準備資料以供應用層使用並定義資料格式的表現方式
  - 資料轉譯、加密和壓縮。
- 工作階段層
  - 這層負責處理開啟和關閉兩個裝置之間的通訊
  - 確保工作階段保持足夠長的開啟時間以傳輸所有進行交換的資料



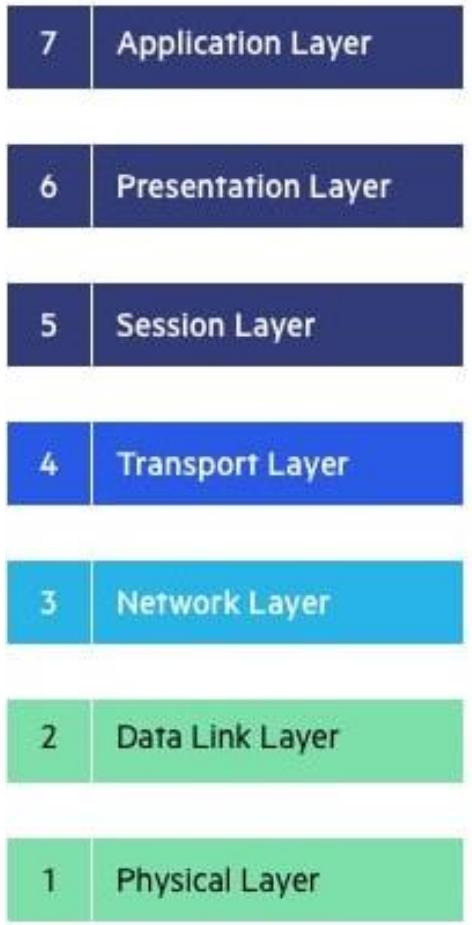
OSI 網路模型7層架構

- 傳輸層
  - 這層負責處理兩個裝置之間的端對端通訊
  - 從工作階段層取用資料，並在傳送至網路層之前分解為稱為"區段"的區塊
  - 接收裝置上的傳輸層負責將區段重組為工作階段層可以取用的資料
- 網路層
  - 這層負責促成兩個不同網路之間的資料傳輸
  - 在傳送者的裝置中將傳輸層中的"區段"分解為較小的單元(封包)
  - 接收端重新組裝這些封包



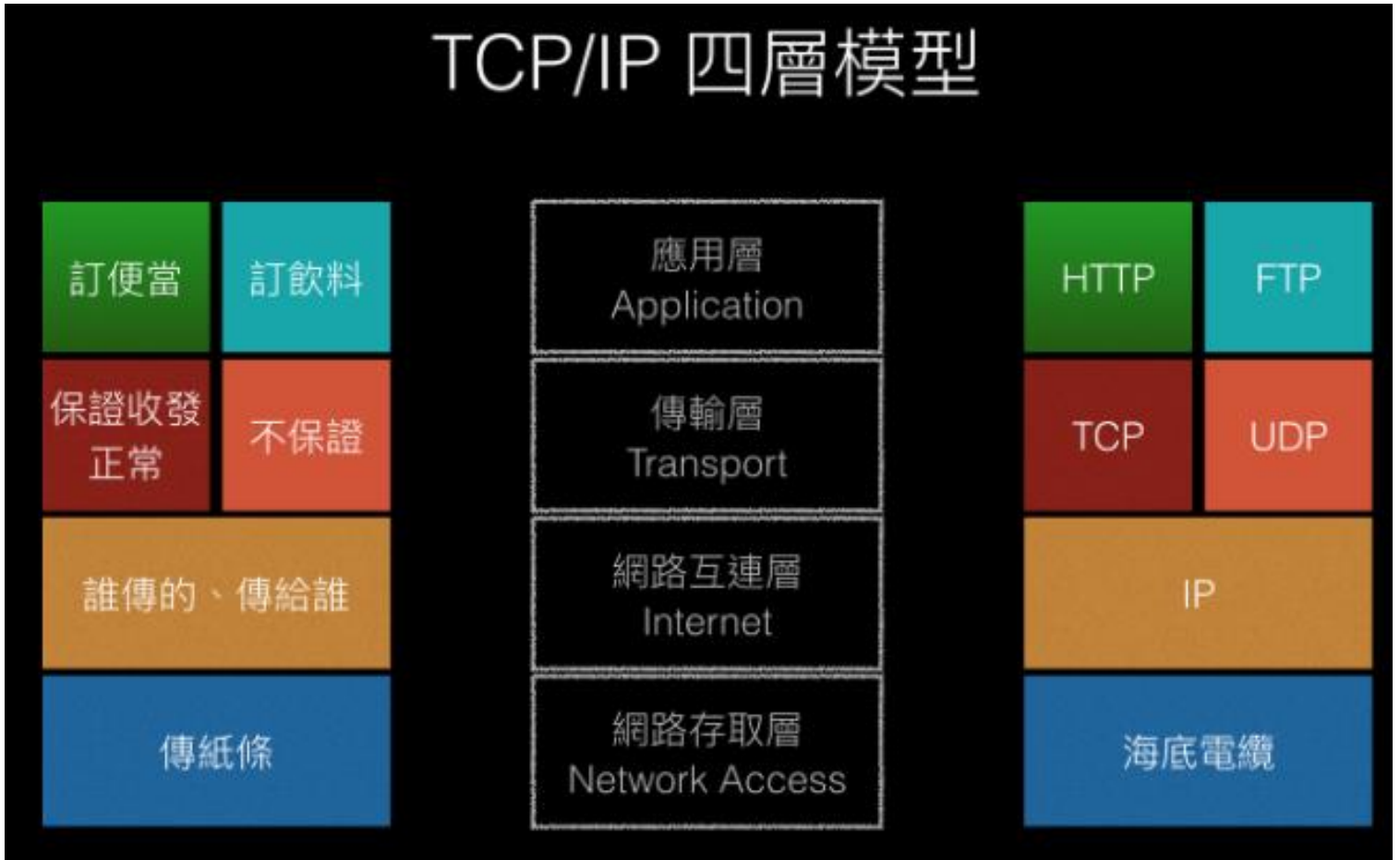
OSI 網路模型7層架構

- 資料連結層
  - 將網路層的封包分割成更小單位訊框 (Frame)
  - 負責網路內的流量控制以及錯誤控制
- 實體層
  - 這層負責網絡節點之間的物理有線或無線連接
  - 資料進一步轉換為 bit stream ，即為一連串的 0 與 1 字串，並轉換為傳輸介質所能傳輸的信號格式



TCP/IP 四層模型架構

› 現如今， 網際網路泛指以TCP/IP為主之通訊協定所架設而成之網路



全球資訊網 (World Wide Web)

- 全球資訊網是檔案、圖片、多媒體和其他資源的全球集合，可以理解為網際網路的一項服務，透過網際網路存取。
- 使用統一資源標誌符標識(URL)
  - 提供了一個全球命名標識系統，象徵性地標識服務、網頁伺服器、資料庫以及提供的檔案和資源
- 超文字傳輸協定(HTTP)
  - 全球資訊網的主要存取協定，全球資訊網的服務使用HTTP在軟體系統之間進行通訊和資料傳輸
- 超文字組成的系統，定義在超文字標記語言(HTML)內
  - 整體透過許多超連結互相連接，便於在資源之間導航

Web的組成要件

- 資源(Resource)
  - 嵌入在網頁上的文字、檔案、多媒體、互動式內容等
- 資源標識符（超連結Hyper link）
  - 為字符串，表示可能包含的通用地址
  - 例如https://ai.nutc.edu.tw/
- 傳輸協議(Transfer Protocol)
  - 規範瀏覽器之間的溝通方式
  - 例如http/https

使用統一資源標誌符標識(URL)

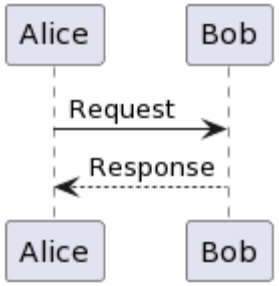
- 網路上的所有資源都是藉由一個URL來定位並存取
- 最早的URL為一長串的IP數字組成
- 後來演變成使用較容易識別的網域名稱以及網域名稱伺服器(DNS)來轉換IP位置並提供網域名稱

使用統一資源標誌符標識(URL)

- URL 由三部分組成
  - 安全協定 (https, ftp)
  - 網域名稱 (www.domain.com)
  - 文件路徑 (/directory/file.html)
- Example:
  - https://en.wikipedia.org/wiki/URL

安全協定　　網域名稱　　文件路徑

超文字傳輸協定(HTTP)



- 規範了客戶端請求與伺服器回應的標準，實際上是藉由 TCP 作為資料的傳輸方式。
- 例如使用者送出了一個請求，資料透過 TCP 協定傳遞給伺服器，並等待伺服器回應；然而這個一來一往的傳輸過程，資料都是 明文傳送。
- HTTPS - 加密過後的HTTP

超文字傳輸協定(HTTP)

- 使用者請求
  - GET: 向指定的資源發出「顯示」請求
  - HEAD: 與GET方法一樣，都是向伺服器發出指定資源的請求。只不過伺服器將不傳回資源的本文部份。
  - POST: 向指定資源提交資料，請求伺服器進行處理（例如提交表單或者上傳檔案）。
  - PUT: 向指定資源位置上傳其最新內容，若內容不存在則新增。

超文字傳輸協定(HTTP)

- 伺服器回應
  - 1XX: 訊息類 (收到請求，請求者繼續執行操作)
  - 2XX: 成功類 (操作被成功接受並處理)，例如：200 成功回應
  - 3XX: 重定向類 (需進一步操作才能完成)，例如：301 成功轉向
  - 4XX: 客戶端錯誤類 (請求語法錯誤或無法完成請求)，例如：404 找不到資源
  - 5XX: 伺服器錯誤類 (後端的問題)，例如：500 伺服器錯誤

標準通用標記式語言 (Standard Generalized Markup Language, SGML)

- 由IBM 在 1960 年代基於通用標記式語言所開發的，是一種將文字以及文字相關的其他資訊結合起來，展現出關於該網頁結構和資料的電腦文字編碼
- SGML是標記式語言的元語言，甚至可以定義不必採用< >的常規方式。由於它的複雜，因而難以普及，同時也是一個國際標準 (ISO 8879:1986)
- 後來的HTML和XML都是由這個語言延伸而來

SGML標準通用標記式語言

- SGML文件包含三個部分
  - 宣告(Declaration)：指定哪些字符和分隔符可能出現在應用程序
  - 文件類型定義(Document Type Definition)：定義標記結構的語法
  - 文件標示：加上標籤處理過後的實際文本
- SGML與HTML最大的不同在於SGML中並沒有定義資料顯示格式的資訊，例如文字的字型、大小與格式，但標籤可以定義出文件的架構

宣告

```
<!SGML  "ISO 8879:1986 (WWW)"
    --
         SGML Declaration for HyperText Markup Language version HTML 4

         With support for the first 17 planes of ISO 10646 and
         increased limits for tag and literal lengths etc.
    --

    CHARSET
          BASESET  "ISO Registration Number 177//CHARSET
                    ISO/IEC 10646-1:1993 UCS-4 with
                    implementation level 3//ESC 2/5 2/15 4/6"
         DESCSET 0       9       UNUSED
                 9       2       9
                 11      2       UNUSED
                 13      1       13
                 14      18      UNUSED
                 32      95      32
                 127     1       UNUSED
                 128     32      UNUSED
                 160     55136   160
                 55296   2048    UNUSED  -- SURROGATES --
                 57344   1056768 57344

CAPACITY        SGMLREF
                TOTALCAP        150000
                GRPCAP          150000
                ENTCAP          150000
```

文件類型定義(DTD)

- DTD有四個組成如下：
- 元素（Elements）: 定義
  - <!ELEMENT 元素名稱 元素內容>
- 屬性（Attribute）
  - <!ATTLIST 元素名稱、屬性名稱、屬性值型態、屬性的內定值>
- 實體（Entities）
  - <!ENTITY 實體名稱　實體內容>
- 注釋（Comments）
  - <!-- 註解內容 -->

可延伸標記式語言 (Extensible Markup Language, XML)

- XML是從標準通用標記式語言（SGML）中簡化修改出來的
- XML被提出是為了有一個更中立的方式，讓客戶端自行決定要如何消化、呈現從伺服器端所提供的資訊。
- XML從1995年開始有其雛形，並向W3C（全球資訊網聯盟）提案，而在1998年二月發佈為W3C的標準（XML1.0）
- XML設計是用來傳送和攜帶資料資訊，不用於表現和展示資料

XML文件的組成

- 文件宣告(Declaration):定義XML文件的版本和使用的字碼集
- 標籤(Tag)：XML能夠自己定義標籤，一個標籤是用來標示文件的部分內容，例如：標籤<code>、<title>和<price>等，標籤分為開頭標籤<code>和結尾標籤</code>。
- 元素(Element)：XML元素為整個文件的主要架構，元素的本身可以是標籤加上文字內容，或是元素內包含有其它的元素，元素是一個完整的項目，它包含標籤、屬性、開始標籤和結尾標籤內的文字內容和結尾標籤。

可延伸標記式語言 (Extensible Markup Language, XML)

- 文件宣告
  - 定義XML文件的版本和使用的字碼集
- 根標籤
  - 定義樹狀結構的根節點
- 元素
  - 根元素下的子元素，也可定義子元素底下的子元素

```
01: <?xml version="1.0" encoding="Big5"?>
02: <!--網頁製作徹底研究系列-->
03: <booklist>
04:   <book>
05:     <code>F8915</code>
06:     <title>ASP網頁製作徹底研究</title>
07:     <authorlist>
08:       <author>陳會安</author>
09:     </authorlist>
10:     <price>580</price>
11:   </book>
12:   <book>
13:     <code>F8916</code>
14:     <title>ASP與IIS 4/5網站架設徹底研究</title>
15:     <authorlist>
16:       <author>陳會安</author>
17:     </authorlist>
18:     <price>550</price>
19:   </book>
20: </booklist>
```

元素

超文本標記語言 (HyperText Markup Language, HTML)

- HTML是一種基礎技術，常與CSS、JavaScript一起被眾多網站用於設計網頁、網頁應用程式以及行動應用程式的使用者介面
- HTML 的目的是呈現和顯示資料， XML 則是攜帶和傳輸資料。
- HTML 具有預先定義的標籤，但使用者可以在 XML 中建立和定義自己的標籤。

HTML 發展背景

- HTML 最初是由 Tim Berners-Lee 在歐洲核子研究中心 (CERN) 開發的以SGML為基礎規範的一個應用程式，並因 Mosaic 瀏覽器而流行，1993年中期網際網路工程任務組（IETF）發布首個HTML規範的提案。
- HTML 2.0
  - IETF建立一個HTML工作群組，並在1995年完成"HTML 2.0"，並追加了表單、表格等規範
- 1996起，HTML的規範就由全球資訊網協會（W3C）來管理和維護

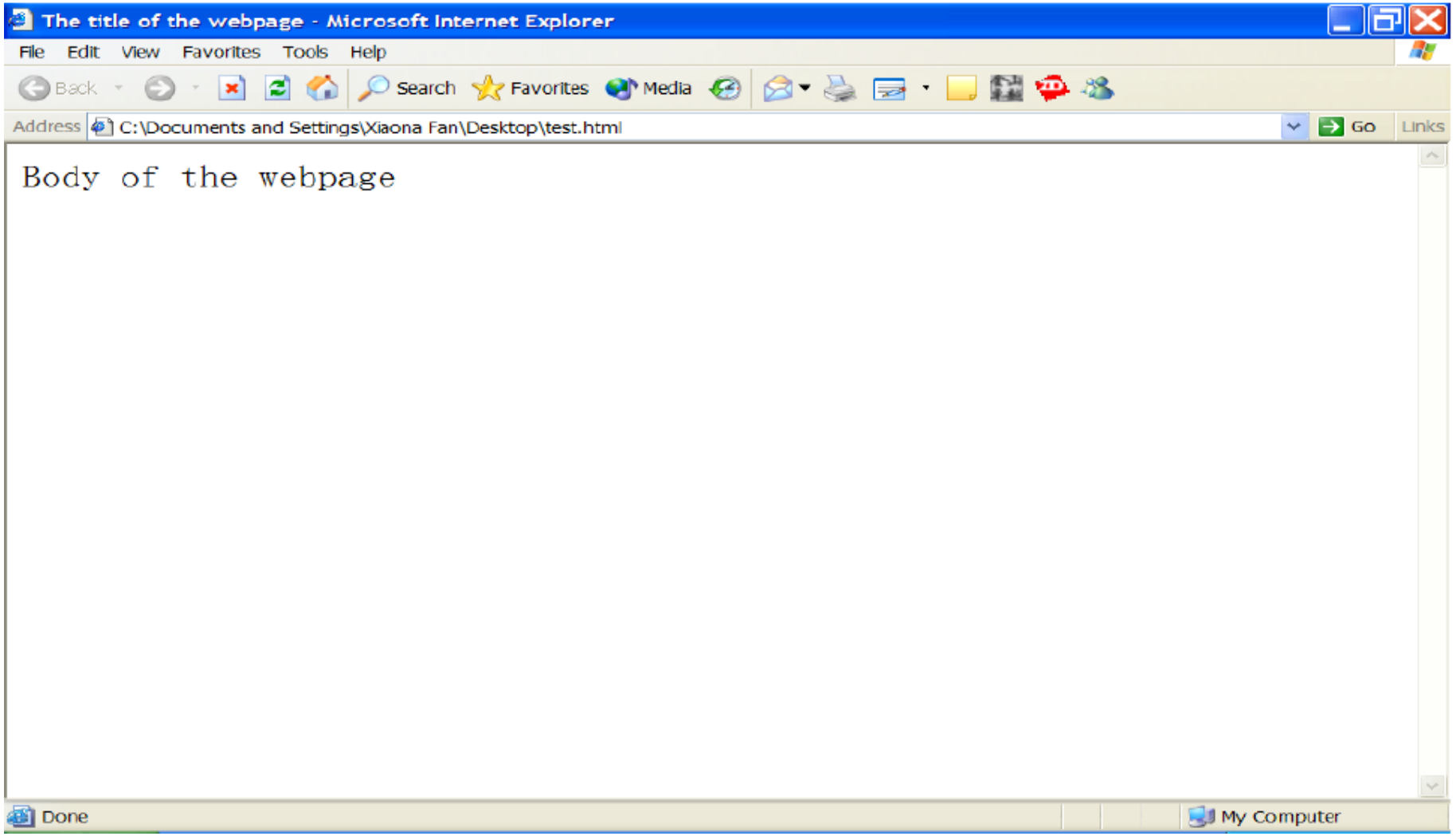
HTML 發展背景

- HTML 3
  - 1997年初HTML 3.2作為W3C推薦標準發布。這是首個完全由W3C開發並標準化的版本，HTML 3.2完全去除數學公式，協調各種專有擴充，並採用網景設計的大多數視覺標記標籤。
- HTML 4
  - 1997年底HTML 4.0作為W3C推薦標準發布。它提供三種變化：
  - 嚴格，過時的元素被禁止。
  - 過渡，過時的元素被允許。
  - 框架集，大多只與框架相關的元素被允許。
- 「XHTML」(eXtensibleHyperTextMarkup Language)
  - 2000年初， W3C所制定用來取代HTML 4.0版的下一個世代HTML
  - 結合XML和HTML4.0版的標籤

HTML 發展背景

- HTML5是HTML最新的修訂版本，由全球資訊網協會（W3C）於2014年10月完成標準制定
- 廣義論及HTML5時，實際指的是包括HTML、CSS和JavaScript在內的一套技術組合。它希望能夠減少網頁瀏覽器對於需要外掛程式的豐富性網路應用服務（Plug-in-Based Rich Internet Application，RIA），並且提供更多能有效加強網路應用的標準集
- 一些過時的HTML 4.01標記將取消，其中包括純粹用作顯示效果的標記，例如<font>和 ，因為它們已經被CSS取代，還有一些透過DOM的網路行為。
- HTML5提供了一些新的元素和屬性，反映典型的現代用法網站。

一個簡單的HTML網站範例



HTML範例程式碼

```
<HTML>
    <HEAD>
    <TITLE>The title of the webpage</TITLE>
    </HEAD>
    <BODY>  <P>Body of the webpage
    </BODY>
</HTML>
```

HTML基礎架構

- 基本上 <head> 裡面的內容都不是給"人"看得，而是給機器運作、搜尋用得。
- 主要放置得標籤用來告訴搜尋引擎，這個網頁有什麼樣的內容、控制網頁與外部程式碼的連結、定義網頁使用的樣式等等。
- HTML5常用的標籤有 <title>、<meta>、<link>、<script>、<style>、<base> 等等
- 網頁真正會跑給使用者看的東西全部都在 <body>

網路爬蟲

- 網路爬蟲是一個透過程式「自動抓取」網站資料的過程，在這資訊爆炸的時代中，資料的收集是相當重要的工作項目之一，但如果透過人工的方式來收集網站資料，效率低之外也會花費掉非常多的時間
- 資料的收集與整理這份工作，可以透過網路爬蟲來協助，我們只要先制定好規則，網路爬蟲就可以自動依照這規則收集和擷取資料並整理出我們所需的格式
  - Excel、CSV等

網路爬蟲的應用

- 找飯店，Trivago!
- Skyscanner 機票搜尋
- 股票應用程式
- 美食推薦應用

網路爬蟲的原理

- 請求網頁內容
  - 網路爬蟲進行的第一步驟都是向目標網站請求特定網址（URL）的內容
- 抓取所需資料
  - 伺服器返回應網頁的 HTML文件後，在此步驟，網路爬蟲主要是將 HTML 文件做「解析」並「取出」所需的資料
- 儲存資料
  - 將取出的資料儲存在 CSV 檔案、Excel 表或是資料庫當中

網路爬蟲合法嗎?

- 透過網路爬蟲每天自動到別人的網站中抓取內容，這時你可能會開始思考一個問題，這樣可以嗎？
- 取決於如何抓取以及怎麼使用抓取到的資料
  - 遵守 robots.txt 的規範
  - 不造成網站伺服器的負擔
- 確認網站是否有提供 API，如有提供API可以直接使用API所定義的程式語法取得資料

Robots.txt

- 通常Robots.txt都在根目錄下，例如
  www.yahoo.com/Robots.txt
  www.google.com/Robots.txt

yahoo

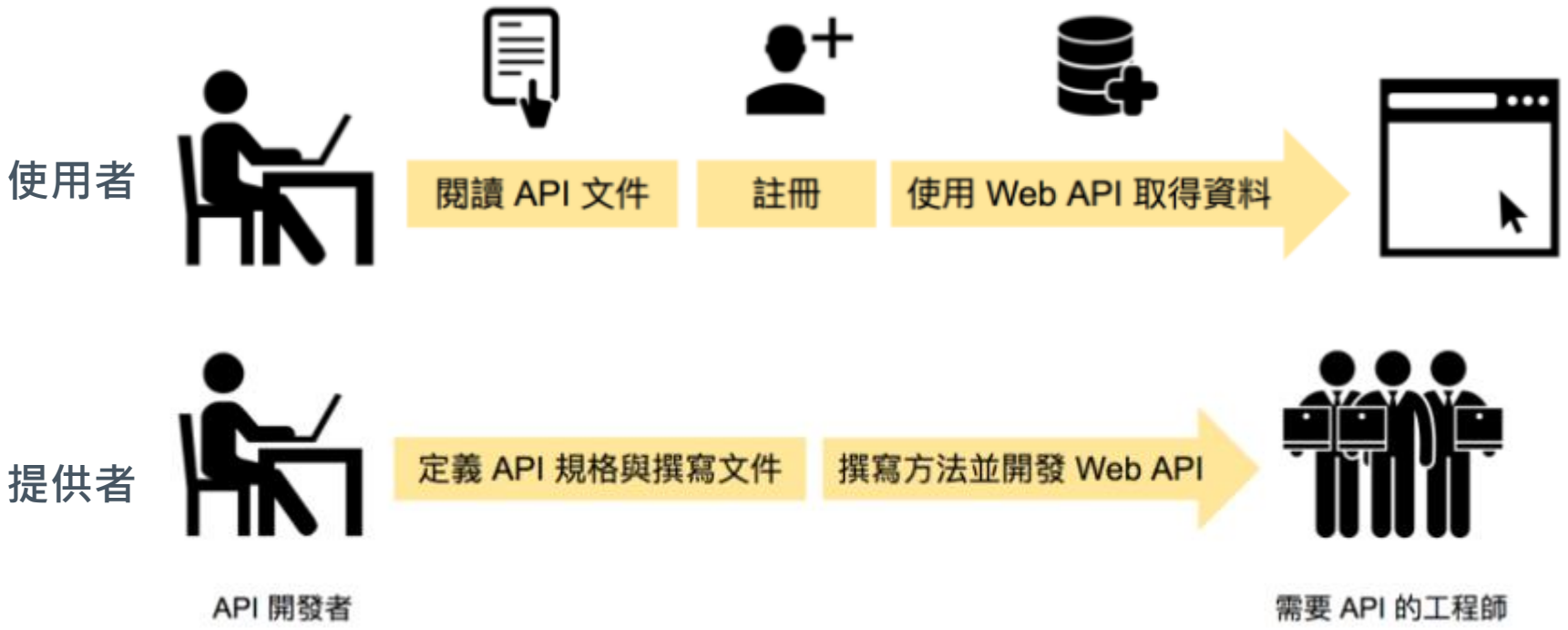
```
User-agent: *
Disallow: /p/
Disallow: /r/
Disallow: /bin/
Disallow: /caas/
Disallow: /blank.html
Disallow: /includes/
Disallow: /_td_api
Disallow: /tdv2_fp
Disallow: /nel_ms
Disallow: /fp_ms
Disallow: /sports_fp_ms
Disallow: /search_ms
Disallow: /_tdpp_api
Disallow: /_remote
Disallow: /_multiremote
Disallow: /_tdhl_api
Disallow: /digest
Disallow: /fpjs
Disallow: /myjs
```

google

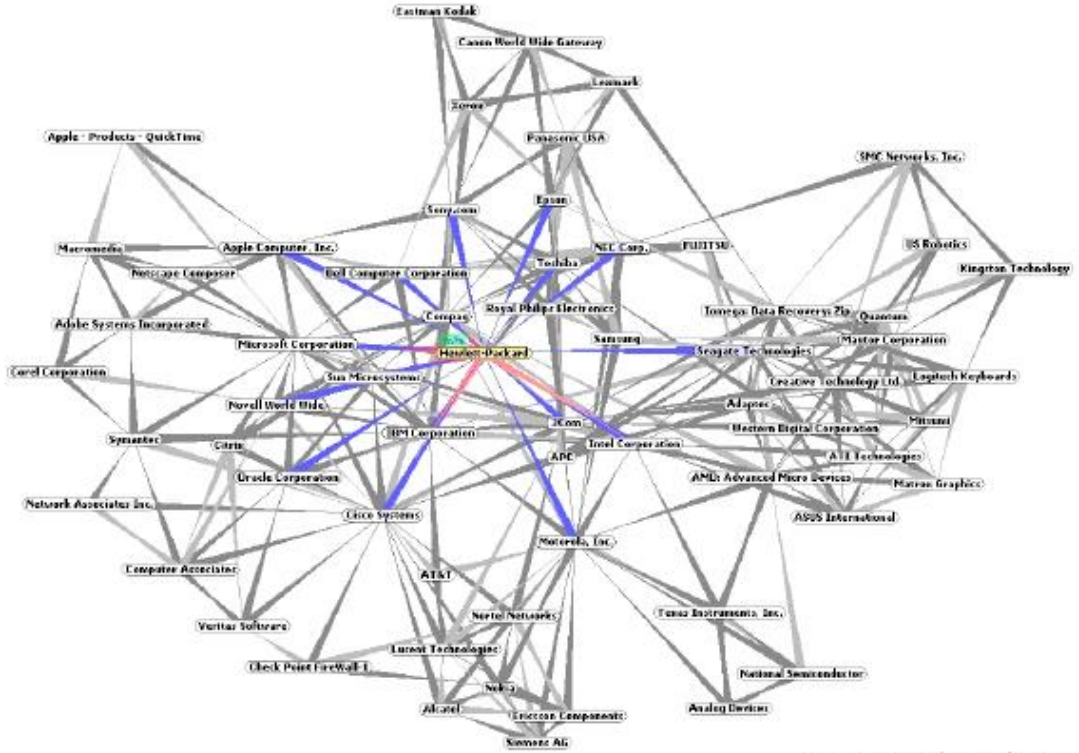
```
User-agent: *
Disallow: /search
Allow: /search/about
Allow: /search/static
Allow: /search/howsearchworks
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*&
Allow: /?hl=*&gws_rd=ssl$
Disallow: /?hl=*&*&gws_rd=ssl
Allow: /?gws_rd=ssl$
Allow: /?pt1=true$
Disallow: /imgres
Disallow: /u/
Disallow: /preferences
Disallow: /setprefs
Disallow: /default
Disallow: /m?
Disallow: /m/
Allow:    /m/finance
Disallow: /wml?
Disallow: /wml/?
Disallow: /wml/search?
Disallow: /xhtml?
Disallow: /xhtml/?
Disallow: /xhtml/search?
```

應用程式介面
Application Programming Interface, API

› 應用程式介面 (API) 是用於打造應用程式軟體的一組副程式定義、協定與工具。一般而言，API 是指各種軟體組件之間一套明確定義的溝通方法



網路圖探勘(Web Graph Mining)



網路圖探勘(Web Graph Mining)

- 網路圖分析
- 網路連結分析
- 網頁重要程度分析
- 異常使用者偵測

網路文本探勘 (Web Text Mining)

搜尋引擎

- 全文檢索
  - 將全部的文字訊息儲存起來
  - 使用者必須詳細的規劃自己的查詢
- 關鍵字查詢
  - 字詞切割
  - 關鍵字定義與比對
- 自然語言處理