



Website

Home

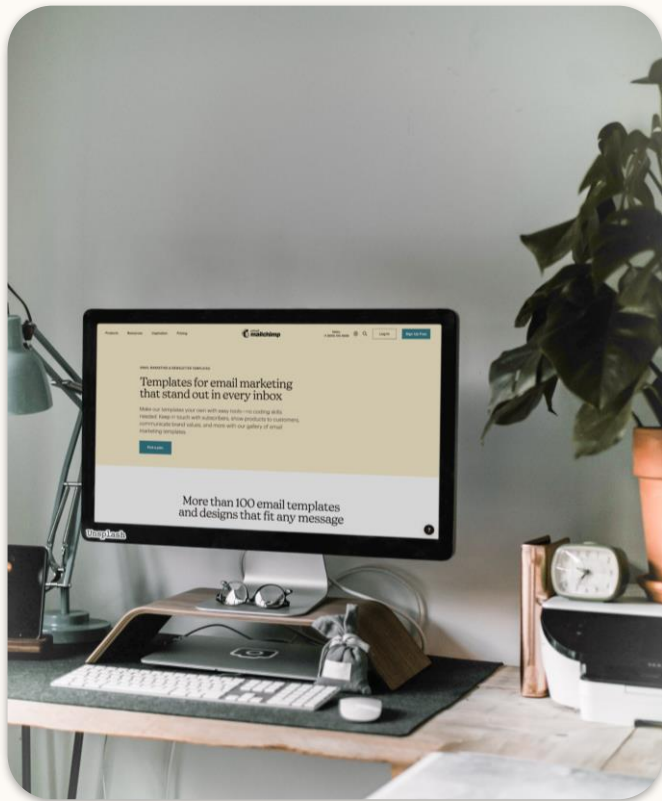
About us

Outline

HTML/CSS

JavaScript

PHP



探索 Web 的世界

# 漸進式 網頁程式 設計課程

桃園市立永豐高中、元智大學資工系學會

[Website](#)[Home](#)[About us](#)[Outline](#)[HTML/CSS](#)[JavaScript](#)[PHP](#)

# About us

- 元智大學
- 資工系二年級
- 專長：C++、Web



桃園市立永豐高中、元智大學資工系學會



Website

Home

About us

Outline

HTML/CSS

JavaScript

PHP



## 第一章 – 基本介紹

# 網站基本概念



Website

Home

About us

Outline

HTML/CSS

JavaScript

PHP

# Lecture Outline

主機之間的溝通

呈現語言  
Html、CSS

呈現軟體

連線地址和Ports

前後端程式語言  
JavaScript、PHP

範例及實作



Website

Home

About us

Outline

HTML/CSS

JavaScript

PHP



## Lesson 1-1

# 主機之間 的雙向溝通



# 簡單來說，網頁是怎麼運作的？

## Phase 1

當我們想要上網時  
在瀏覽器輸入想去的網址  
瀏覽器向伺服器送出請求  
開始等待伺服器回應



## Phase 2

伺服器收到我們的請求  
準備好我們要的內容並回傳

## Phase 3

瀏覽器接收到回應  
依照檔案中的指示排版與執行  
準備就緒時呈現給使用者





## Q：伺服器（Server）是什麼？

- 規模可大可小
- 每臺電腦經過「適當設置」都可成為伺服器
- 儲存必要檔案，並提供服務
- 接收客戶端使用者的請求並做出適當回應





## Q：我的電腦怎麼找到伺服器？

- 主機之間是用IP來認得對方，並非網址
- 網址只是方便我們記憶而發明的
- 當我們輸入網址，透過「DNS」，將網址轉為IP

例如：當我們想要前往Google

輸入Google.com ➡ DNS開始查詢 ➡ 找到142.251.42.238







## Q : IP是什麼?

- 每台裝置只要連上網，都會被分發到一個地址
- 這個地址就是IP，作為網路世界中的身分辨識碼
- IP由四個數字組成，每個數字範圍0~255



實作：找出自己電腦的IP

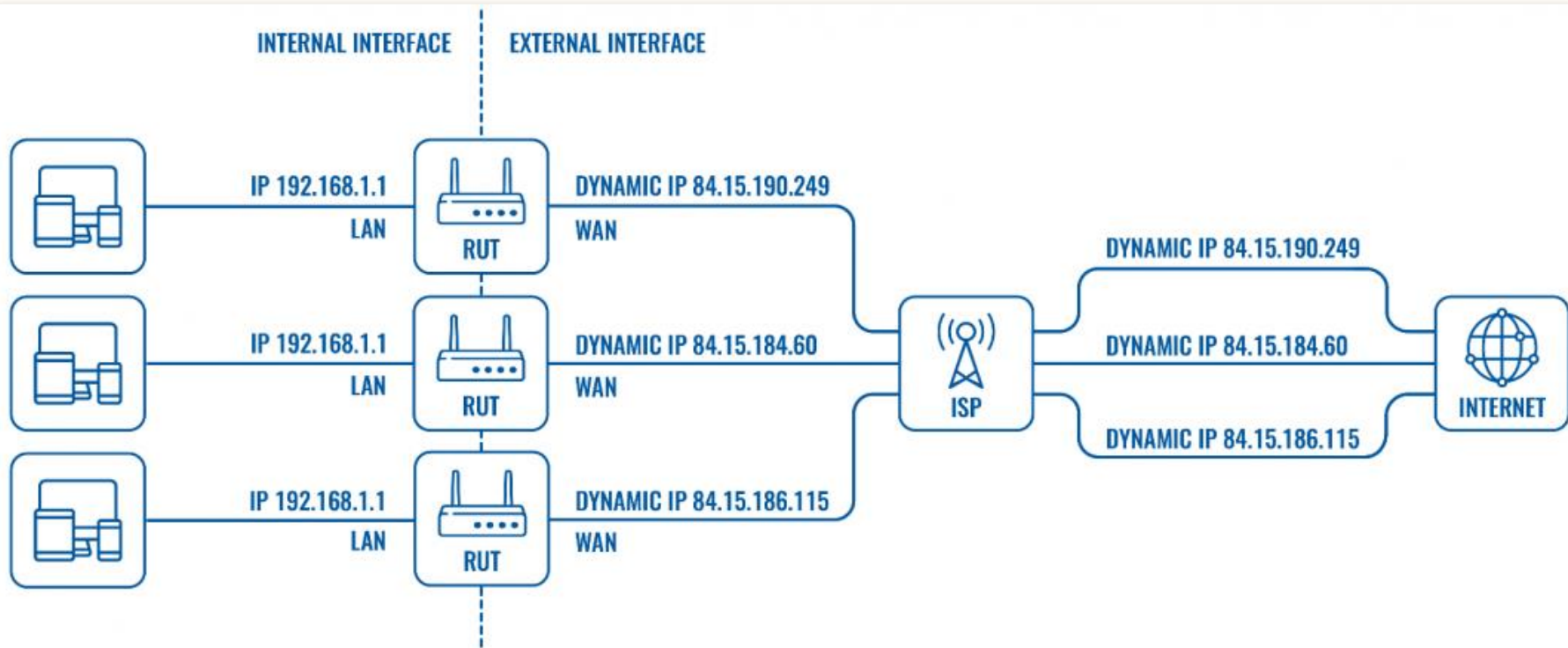




## Q：全球裝置這麼多，IP夠用嗎？

- IP總共有256的四次方個（4,294,967,296個）
- 分成「內網IP」和「對外IP」（見下頁例圖）
  - ➡ 內網IP：僅在同網域有效，不同網域的裝置可以重複IP
  - ➡ 對外IP：不可與他人重複，唯一的識別碼
- IP逐漸用盡，出現IPv6：位址數量更多







# 雙方協調好的溝通方式

- 雙方遵照HTTP ( HyperText Transfer Protocol ) 協定的規範進行溝通
  - HTTP傳送內容時不會經過加密，較不安全
- 為了安全，出現HTTPS ( HTTP Secure )，內容經過加密
- 伺服器使用Port 80 (HTTP)/Port 443 (HTTPS)來接收請求與傳輸
- 常見請求方法：Get (顯示請求) / POST (資料提交)





## Q : Port是甚麼?

- 可以把Port想成每一台主機通往網路世界的大門
- 每部裝置都有65536個Port可供不同服務同時使用
- Port平常是關閉的，有傳輸需求時會被開啟並佔用
- [Well known ports & ephemeral ports](#)





# 經過「適當設置」成為伺服器

1. 開啟 80 port
2. 使用軟體接收HTTP請求並回應



實作：用「XAMPP」完成以上需求



XAMPP



XAMPP將最基本的Apache HTTP伺服器、MySQL資料庫  
以及PHP打包在同一個安裝包中，功能強大，操作簡易



# 實驗：第一次連線

1. 打開記事本，任意輸入一串文字，並存檔為first.html
2. 將檔案放在 C:\xampp\htdocs\ (要對外公開的檔案都放在這裡)
3. 使用剛剛查詢到的IP，邀請同學連線進入



網址為 [http://xxx.xxx.xxx.xxx\(你的IP\)/first.html](http://xxx.xxx.xxx.xxx(你的IP)/first.html)



補充：在本機測試時，可以直接輸入 <http://localhost/>



用HTTP的方式 對「自己」的 80 Port 送出請求

[Website](#)[Home](#)[About us](#)[Outline](#)[HTML/CSS](#)[JavaScript](#)[PHP](#)

# Localhost?

I started to develop a website with HTML, and this is very good

Do you want to see?

Yes, send me the link.

<http://localhost/index.html>

17:28

Your website looks like mine.

It runs fine on my browser

Email me the link

<http://localhost/test1>

< Messages Jimmy Details

@\_codehub\_

Hey bro, after days of hard work, I finally built a website

That's awesome!

Can you check it out??

<http://localhost:3000/hi>

Looks great bro!

Thanks bro!





Website

Home

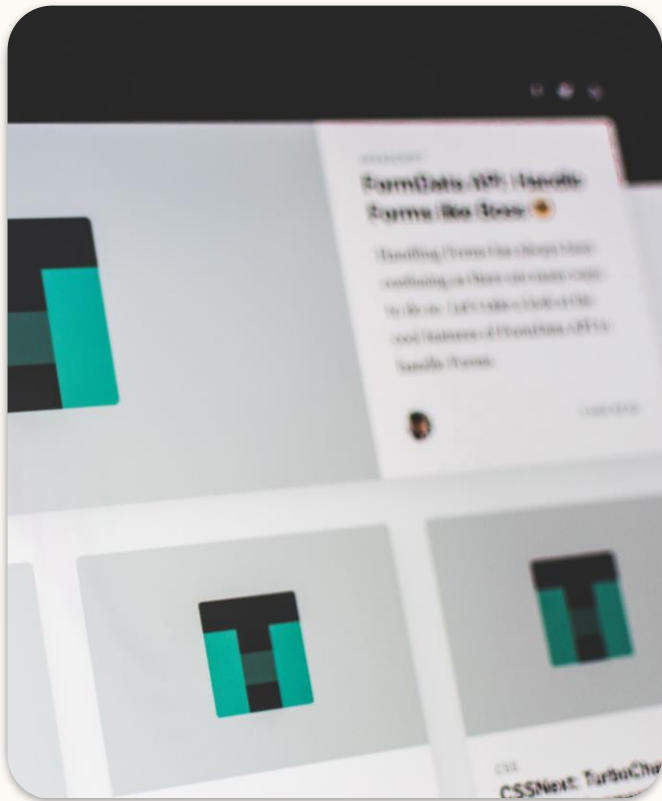
About us

Outline

HTML/CSS

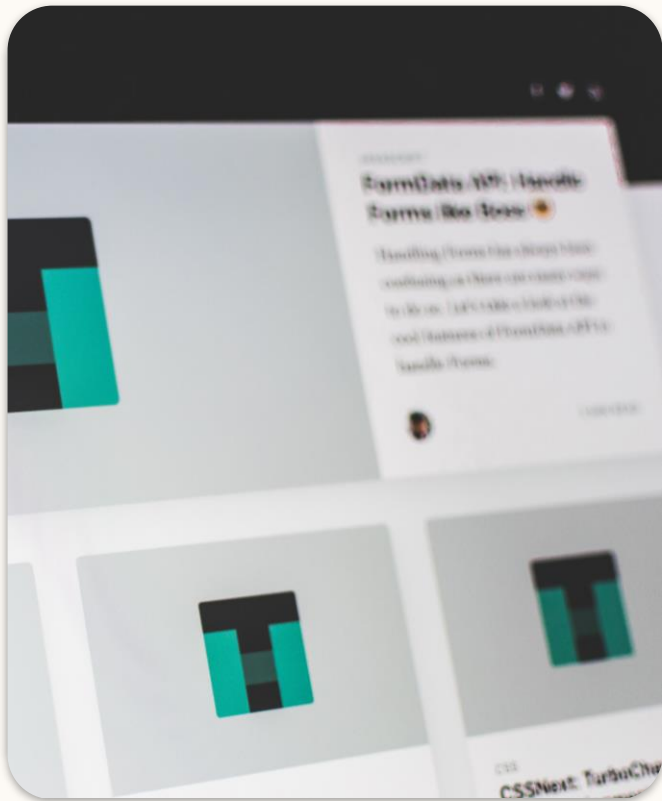
JavaScript

PHP



## Lesson 1-2

# 網頁內容 的組成元素



## Lesson 1-2

經過剛剛的實驗可以知道，  
一般的文字就可以當成網頁顯示。  
但是否有些單調呢？  
要怎麼美化剛剛的頁面？



# 網頁呈現語言：HTML

- HyperText Markup Language
- 是一種「標記語言」（並非程式語言）
- 使用「開始標籤」及「結束標籤」將網頁切成許多區段
- 有各種不同功能的標籤，加入文字以外的元素
  - ➡ 例如：嵌入圖片影片、互動式表單、表格、超連結
- 配合CSS使用，可以進一步指定標籤區段的外觀和風格





# 網頁呈現語言：CSS

- Cascading Style Sheets 階層式樣式表（非程式語言）
- 必須與標記語言搭配使用，用於指定網頁排版及外觀
- 通常以HTML中的 id 和 class 來辨識不同區段或類別
- 也可使用CSS製作簡單的動畫
- 每個瀏覽器可能對相同的CSS有些微不同的詮釋
- 對於靜態網頁來說，HTML與CSS已經足夠

```
a.fs-link {  
    position: relative;  
    padding: 4px 2px;  
    font-weight: bold;  
    color: inherit;  
    background: linear-gradient(to right, #000 49%, #fff 49%, #fff 51%, #000 51%);  
    background-position: 0 0, 0 0;  
    transition: background-position 0.5s;  
  
    &:hover {  
        background-position: 50% 0, 50% 0;  
    }  
}
```



# HTML & CSS 不是程式語言！





# 前端程式語言：JavaScript

- 可增加網頁互動性，讓網頁變得更加動態
- 為程式語言，可利用變數暫存資料、條件判斷及運算
- 在使用者端瀏覽器中執行，伺服器僅負責提供檔案
- 有許多開源的函式庫，讓JavaScript更加實用
  - ➔ 例如：Jquery、React、Vue





# 後端程式語言：PHP

- 在伺服器端執行，將執行後的輸出傳送到前端顯示
- 使用者是無法看到PHP原始碼的（只會看到執行結果）
- 可以接收前端POST回來的資訊並進行處理
- 可以連接MySQL資料庫進行資料儲存及查詢



常見用法：用於帳號密碼註冊及登入



```
*  
* @var boolean  
*/  
define('PSI_INTERNAL_XML'  
  
if (version_compare("5.2"  
    die("PHP 5.2 or greater  
}  
if (!extension_loaded("p  
    die("phpSysInfo req  
        properly.");  
}  
  
require_once APP_ROOT.  
  
// Load configuration  
require_once APP_ROOT  
  
if (!defined('PSI_CO  
    $tpl = new Templ  
    echo $tpl->fetch  
    die();
```





Website

Home

About us

Outline

HTML/CSS

JavaScript

PHP

# 還有很多不同的後端程式語言

## Top backend languages in 2022



JavaScript



Python



PHP



Golang



C#



Rust



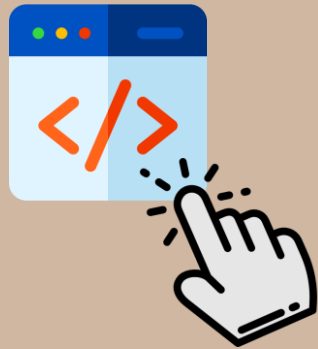
Typescript





# 實驗：查看網頁組成元素

1. 進入任何一個網站
2. 按下F12開啟開發者工具
3. 點進sources欄位，觀察組成該網頁的元素有哪些
4. 操作網頁的同時，觀察是否有變化



[Website](#)[Home](#)[About us](#)[Outline](#)[HTML/CSS](#)[JavaScript](#)[PHP](#)

# 準備開始創作第一個網頁！



編輯工具：請下載 [Visual Studio Code](#) 或 [Sublime text](#)

