



# Web Programming Introduction

海大資工 馬尚彬

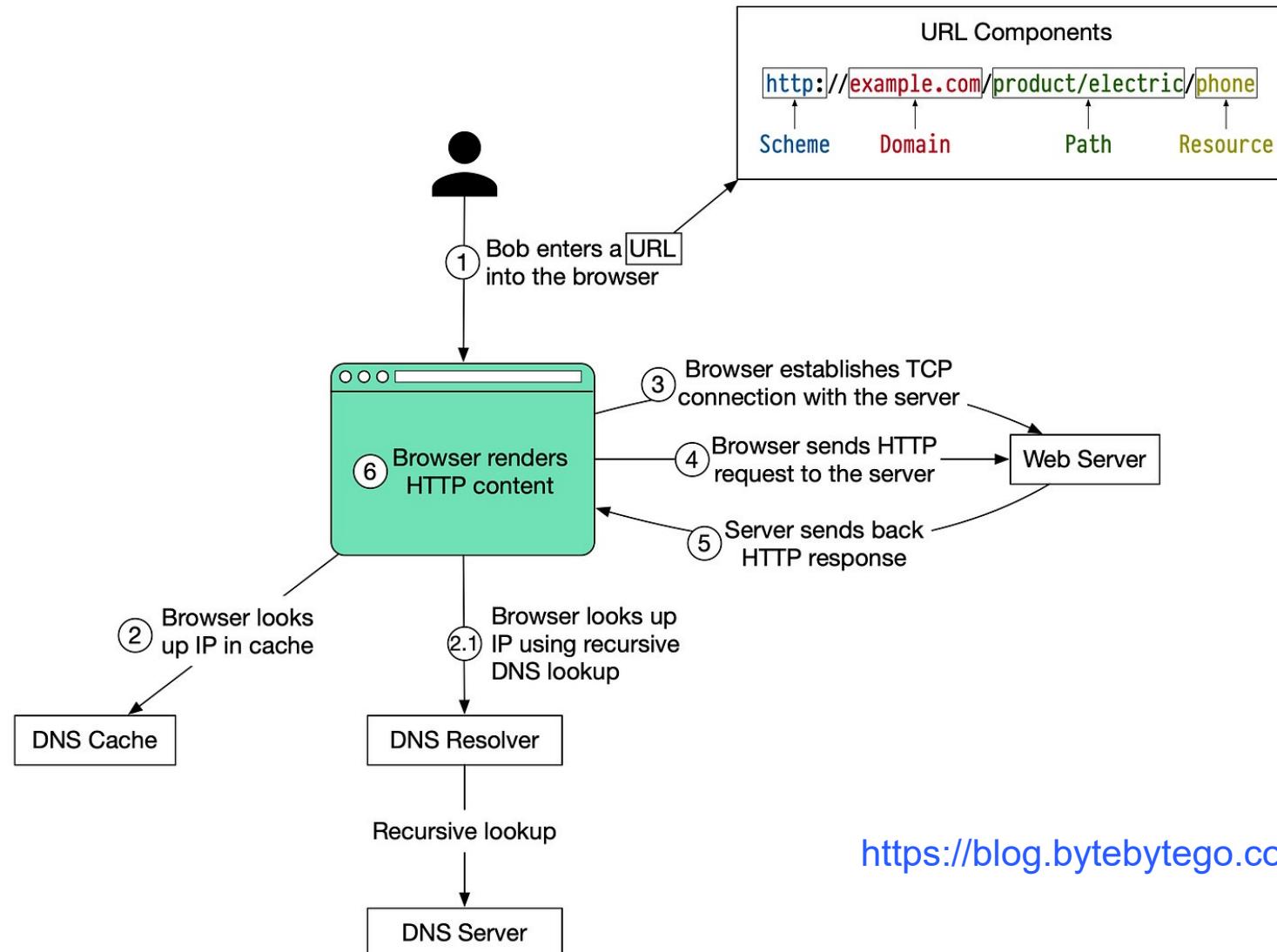
2025

# World Wide Web (WWW)

- 由Tim Berners-Lee創立: <https://zh.wikipedia.org/zh-tw/蒂姆·伯纳斯-李>
- 一個Internet上用來存取鏈結文件(linked documents)的一個架構
- 最一開始創立的目標以文件分享為主，再配合超連結的觀念
- 從靜態的內容，到動態的內容與多媒體

# Browser and Web Server

| What happens when you type a URL into your browser?



# Web Communication Protocols<sub>1</sub>

- ▶ A Web page is identified by a unique address called the URL.
- ▶ Hypertext Transfer Protocol (HTTP) manages the hypertext links that are used to navigate the Web
  - HTTP是Browser與Web Server溝通的語言
- ▶ HTTP/1.1 was standardized in RFC 2616 in 1999.
  - It introduced persistent connections by default, chunked transfer encoding, and additional caching mechanisms.

# Web Communication Protocols<sub>2</sub>

- ▶ HTTP/2 was published as a Proposed Standard on February 17, 2015.
  - It introduced multiplexing, header compression, and server push to improve performance over HTTP/1.1.
  - <https://w3techs.com/technologies/details/ce-http2>
- ▶ HTTP/3 was officially standardized in June 2022 as RFC 9114, based on the QUIC (Quick UDP Internet Connections) transport protocol.
  - It enhances reliability and reduces latency by using UDP instead of TCP.
  - <https://w3techs.com/technologies/details/ce-http3>

# HTML Documents

- Web pages are created using Hypertext Markup Language (**HTML**)
- Web pages are commonly referred to as **HTML pages** or **documents**
- A **markup language** is a set of characters or symbols that define a document's logical structure
- **HTML** has evolved into a language that defines how elements should appear in a Web browser
- 你是否曾經「檢視原始檔」？

# Well-Formed Web Pages

- ▶ HTML became an Internet standard in 1993 with the release of version 1.0.
- ▶ HTML5
  - As of October 2014 this is the final and complete fifth revision of the HTML standard of the World Wide Web Consortium (W3C).
  - <video>, <audio>, and <canvas> elements
    - include and handle multimedia and graphical content on the web without having to resort to proprietary plugins and APIs.
  - <section>, <article>, <header>, and <nav>
    - enrich the semantic content of documents.



# Cascading Style Sheets

- ▶ A single piece of CSS formatting information, such as text alignment, is referred to as a style
- ▶ The term cascading refers to the ability for Web pages to use CSS information from more than one source

HTML只管文件內容，CSS負責呈現！  
(有質感的Website都靠好的CSS!)



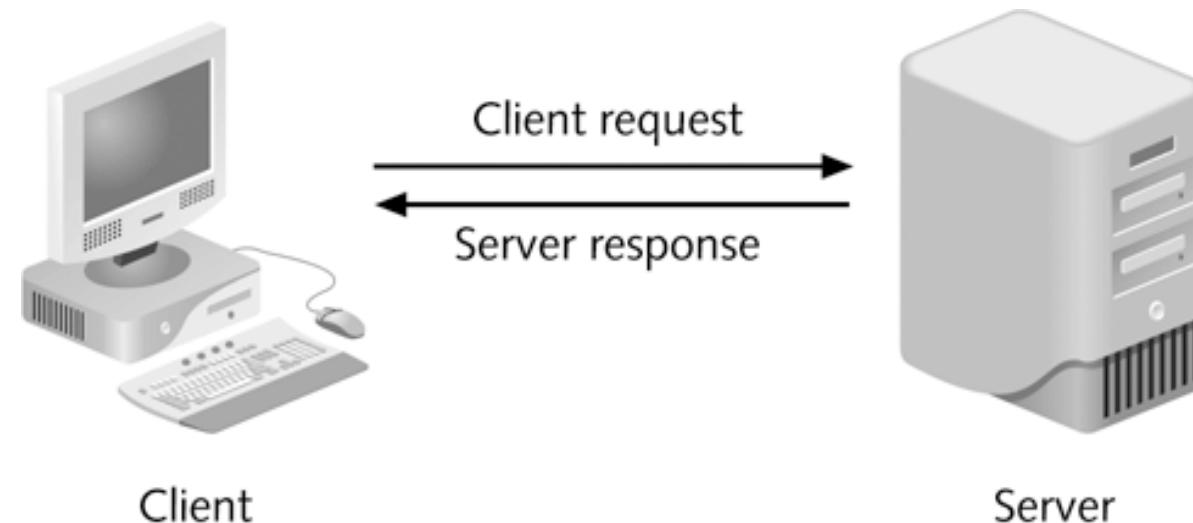
[https://www.reddit.com/r/ProgrammerHumor/comments/68x8bm/with\\_css\\_vs\\_without\\_css/](https://www.reddit.com/r/ProgrammerHumor/comments/68x8bm/with_css_vs_without_css/)

# Client/Server Architecture<sub>1</sub>

- ▶ Server ("backend"):
  - Fulfils a request for information by managing the request or serving the requested information to the client.
  - Responsible for data storage and management.
- ▶ A system consisting of a client and a server is known as a **two-tier system**.

# Client/Server Architecture<sub>2</sub>

- ▶ Client ("front-end"):
  - Presents an interface to the user
  - Gathers information from the user, submits it to a server, then receives, formats, and presents the results returned from the server



# Client-Side Scripting

- ▶ Client-side scripting is a language that runs on a local browser (on the client tier) instead of on a Web server (on the processing tier).
  - Libraries: jQuery, d3.js (c3.js), Leaflet, babylon.js, etc.
  - Frameworks: reactJS, AngularJS, Vue.js, etc.

# JavaScript

- ▶ JavaScript is a client-side scripting language that allows Web page authors to develop interactive Web pages and sites
  - Used in most Web browsers including Chrome, Edge, Firefox and Safari.
- ▶ JavaScript allows you to:
  - Turn static Web pages into applications such as games or calculators.
  - Change the contents of a Web page after a browser has rendered it.
  - Create visual effects such as animation.

要讓前端網頁能動，大部分靠JavaScript!

# Why Backend?<sub>1</sub>

- ▶ Data Management
  - Connect and interact with databases
  - Ensure data integrity and security
- ▶ System Integration
  - Bridge modern web apps with legacy systems
  - Facilitate API connections with third-party services
- ▶ User Interaction
  - Enable real-time communication between users
  - Manage user authentication and authorization

# Why Backend?<sub>2</sub>

- ▶ Business Logic
  - Implement and protect core application functionality
  - Centralize critical operations for consistency and maintainability
- ▶ Asynchronous Processing
  - Handle background tasks and scheduled jobs
  - Push notifications and updates to users
- ▶ Scalability and Performance
  - Optimize server-side operations for improved response times
  - Manage server resources to handle increasing loads

# Server-Side Scripting

- ▶ **Server-side scripting** refers to a scripting language that is executed from a Web server
- ▶ Common server-side scripting technology:
  - Java: Spring/Spring Boot
  - Python: FastAPI, Flask, Django
  - C#: ASP.Net
  - PHP: Laravel, Lumen, Codeigniter
  - **node.js (JavaScript)**: Express.js, Koa.js, Socket.io

# PHP

- ▶ **Hypertext Preprocessor (PHP)** is a server-side scripting language that is used to develop interactive Web sites.
  - Is easy to learn.
  - Includes object-oriented programming capabilities.
  - Supports many types of databases (MySQL/MariaDB, Oracle, MongoDB, etc.).
  - Well-Known frameworks: Laravel, Lumen, Codeigniter

# Node.js

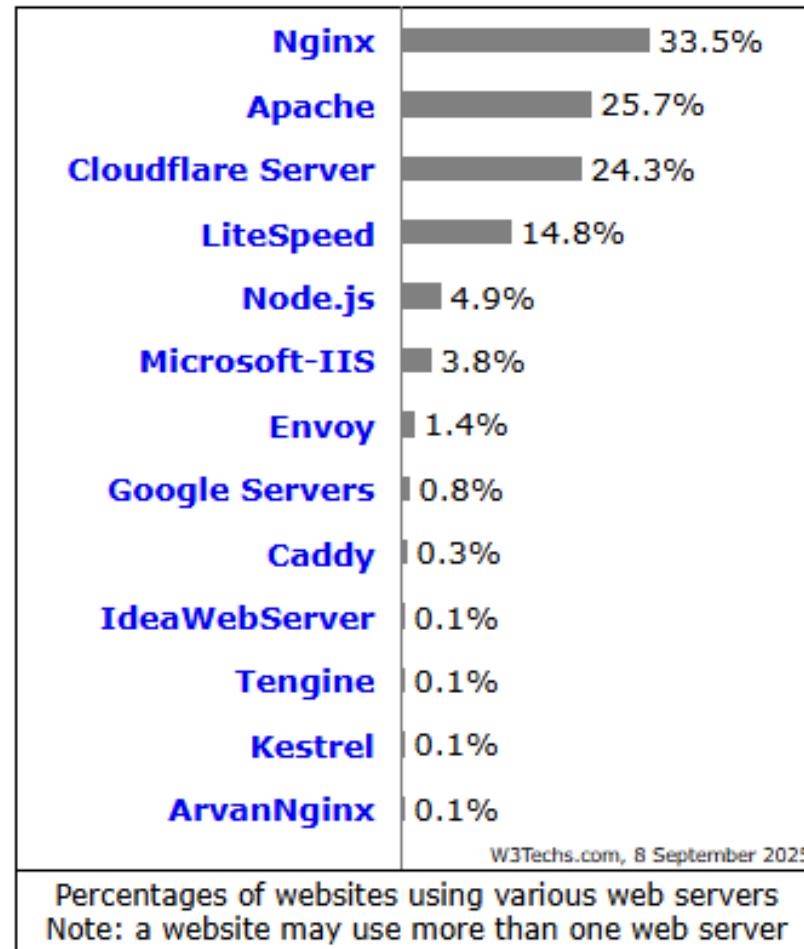
- ▶ **Node.js** is an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side.
  - Enables JavaScript to be used for server-side scripting, and runs scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.
  - Has an event-driven architecture capable of non-blocking, asynchronous I/O.
  - Well-Known frameworks: Express.js, Koa.js, Socket.io

# Browser Market Share

<b>2025</b>	<b><u>Chrome</u></b>	<b><u>Edge</u></b>	<b><u>Firefox</u></b>	<b><u>Safari</u></b>	<b><u>Opera</u></b>
August	80.4 %	11.2 %	3.5 %	3.4 %	0.9 %
July	80.5 %	11.3 %	3.6 %	3.3 %	0.8 %
June	79.8 %	11.4 %	3.6 %	3.7 %	0.9 %
May	79.1 %	11.6 %	3.7 %	4.0 %	1.0 %
April	79.1 %	11.4 %	3.8 %	4.2 %	0.8 %
March	78.5 %	11.6 %	3.8 %	4.2 %	0.8 %
February	78.4 %	11.7 %	3.9 %	4.3 %	0.9 %
January	79.4 %	10.7 %	3.9 %	3.9 %	1.4 %

<https://www.w3schools.com/browsers/default.asp>

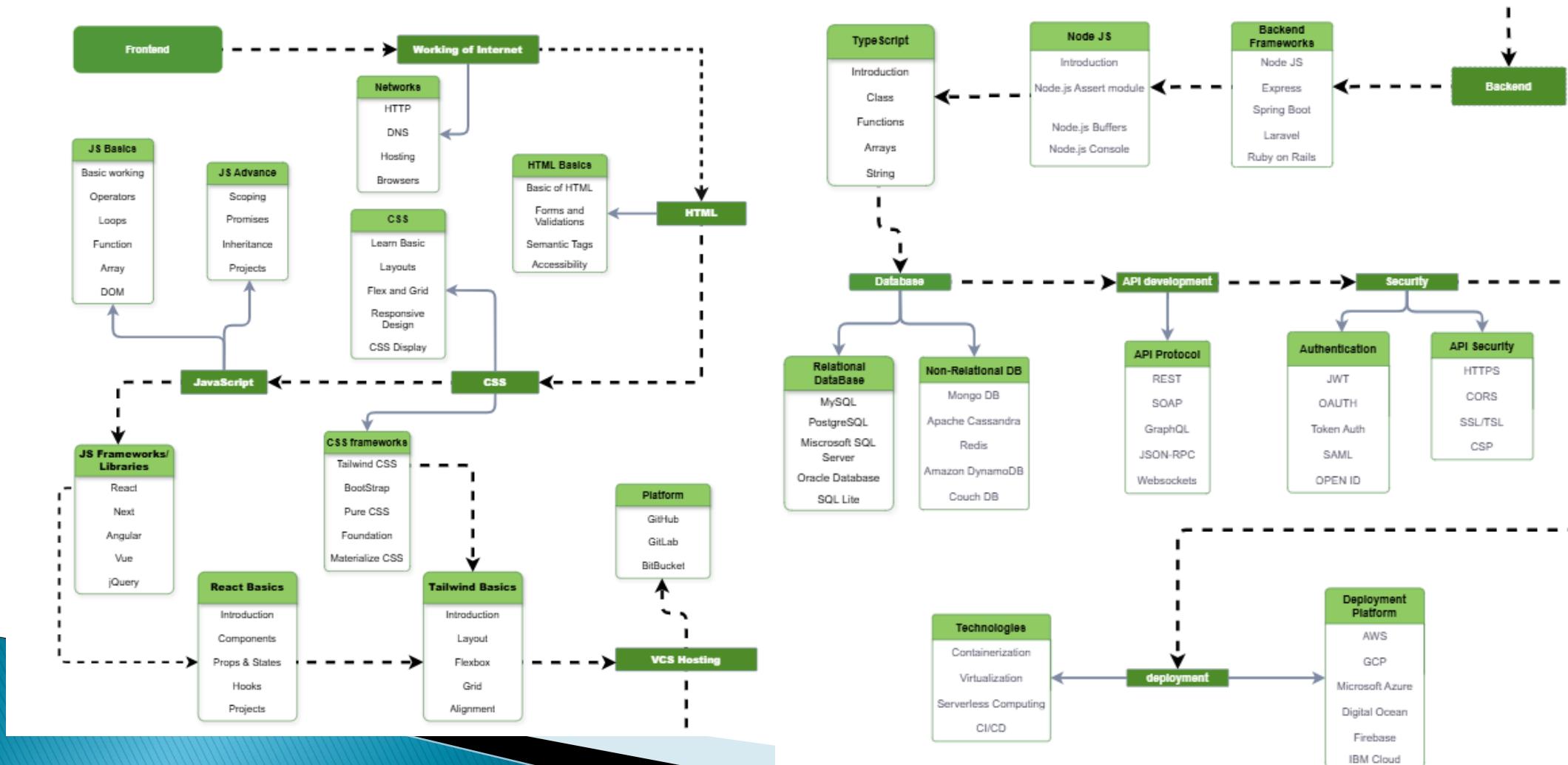
# Percentages of Websites using Various Web Servers



[https://w3techs.com/technologies/overview/web\\_server/all](https://w3techs.com/technologies/overview/web_server/all)

# Full Stack Roadmap

- ▶ <https://www.geeksforgeeks.org/blogs/full-stack-developer-roadmap/>



# 課前問卷

- ▶ <https://forms.gle/TWQ5P1RUtnQiXVMD9> or  
<http://tiny.cc/2025-web-survey>

