



# 網站與前端框架概觀

---

Web 前端框架實戰 | 許季秦 | 113.02.26

# 在學習框架之前，需具備的基本知識



**HTML/CSS**

**JavaScript**

**DOM** ( 文件物件模型 )

**Node.js** 和 **npm** ( **Node** 套件管理器 )

版本控制系統 ( 如 **Cit** )

基本的前端概念

# HTML/CSS

HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are fundamental technologies used to create and design web pages.

# HTML (Hypertext Markup Language)



## ■HTML ( 超文本標記語言 )

- HTML是用於定義網頁結構的標記語言
- 使用一系列的標籤 ( tag ) 來構建網頁各個不同的部分

# CSS (Cascading Style Sheets)



## ■CSS ( 層疊樣式表 )

- CSS用於定義網頁元素的樣式和外觀
- 能夠控制字體、顏色、間距與佈局等方面
- 層疊性(Cascading)表示樣式的繼承性，即樣式可以層層堆疊
- CSS可以與HTML分離，以外部樣式表的形式存在，透過連結(link)引入HTML文件

# JavaScript

是一種用於網頁開發的動態程序語言，它主要用於瀏覽器中的客戶端腳本，可以在網頁上執行交互式的操作。

# JavaScript 基本概念

動態程序  
語言

客戶端程序  
語言

語法和結構

事件驅動

物件導向

非同步編程

跨平台

第三方類別  
庫和框架

- 動態程序語言，表示可以在執行時修改和操作網頁的內容、結構和樣式。

# JavaScript 基本概念

動態程序  
語言

客戶端程序  
語言

語法和結構

事件驅動

物件導向

非同步編程

跨平台

第三方類別  
庫和框架

- 主要用於瀏覽器端，與用戶互動，動態更新頁面，檢驗表單資料，處理事件等。



# JavaScript 基本概念

動態程序  
語言

客戶端程序  
語言

語法和結構

事件驅動

物件導向

非同步編程

跨平台

第三方類別  
庫和框架

- JavaScript 與 HTML 和 CSS 一起工作。
- 可以直接內嵌在HTML中，也可以作為外部程序文件連結到HTML中。

# JavaScript 基本概念

動態程序  
語言

客戶端程序  
語言

語法和結構

事件驅動

物件導向

非同步編程

跨平台

第三方類別  
庫和框架

- 可以響應各種事件，如用戶的點擊、輸入等，並且可以執行相應的程式碼。

# JavaScript 基本概念

動態程序  
語言

客戶端程序  
語言

語法和結構

事件驅動

物件導向

非同步編程

跨平台

第三方類別  
庫和框架

- 一種物件導向的語言。
- 使用物件（ Objects ）來組織程式碼，允許建立和使用自定義物件。

# JavaScript 基本概念

動態程序  
語言

客戶端程序  
語言

語法和結構

事件驅動

物件導向

非同步編程

跨平台

第三方類別  
庫和框架

- 支援非同步編寫程式，例如使用call back 函數、Promise 物件，以及的 async/await 語法，這允許處理非同步操作，如資料的請求。

# JavaScript 基本概念

動態程序  
語言

客戶端程序  
語言

語法和結構

事件驅動

物件導向

非同步編程

跨平台

第三方類別  
庫和框架

- 原本是瀏覽器端的程序語言，但隨著 Node.js 的出現，JavaScript 現在也能在伺服器端執行。

# JavaScript 基本概念

動態程序  
語言

客戶端程序  
語言

語法和結構

事件驅動

物件導向

非同步編程

跨平台

第三方類別  
庫和框架

- 有許多流行的 JavaScript 框架和函式，如React、Angular、Vue等，用於簡化和加速前端開發。

# DOM ( 文件物件模型 )

Vue.js 和 React 都與瀏覽器的 DOM 進行交互，因此對 DOM 操作和事件處理的基本知識會很有幫助。

# DOM ( 文件物件模型 )



- 一種用於表示和操作HTML和XML文檔的程式化的標準。
- 將文件視為一個樹狀結構，其中每個節點代表文件中的一個部分，並且每個節點都是一個物件。



# Node.js npm ( Node 套件管理器 )

Node.js 是一個開源的、跨平台的JavaScript運行環境，它使用了Chrome V8引擎，能夠在伺服器端運行JavaScript代碼。

# Node.js /npm



## ■ Node.js

- 開發者可以使用Node.js來構建後端服務、API、即時應用程式（如聊天應用程式），以及其他各種類型的應用程式。

# Node.js /npm



## ■ Npm

- 是 Node.js 的套件管理器，用於管理和共享JavaScript代碼。
- 使用Node.js開發應用程式時，通常會依賴於各種外部套件和函式庫，這些套件和函庫透過npm進行安裝和管理。
- Vue.js 和 React 的開發環境通常使用 npm 來管理相依項目。

# 版本控制系統（如 Git）

在團隊中協作開發時，版本控制是必不可少的Git 將有助於更有效地管理程式碼版本。

# 版本控制系統（如 Git）



- 版本控制系統（Version Control System，縮寫為VCS）是一種記錄和管理項目中文件變更的系統。
- 允許開發者或團隊追蹤項目的變更，協同工作，並隨時回溯到先前的狀態。
- Git 是目前最廣泛使用的版本控制系統之一。

# 基本的前端概念

瞭解單向資料流、組件化、狀態管理等前端開發的基本概念，這將有助於容易地理解 Vue.js 和 React 的設計理念。

# 基本的前端概念



- 前是指與用戶直接交互的部分，通常是指網頁或應用程式的使用者界面。
- 包含HTML、CSS、Javascript、瀏覽器、DOM、AJAX（非同步JavaScript和XML）、Responsive Web Design、前端框架和函式庫、網頁性能優化以及版本控制系統等。

