

前端工具教學

探索現代前端開發的核心工具
從CSS框架到JavaScript框架的完整指南



</> Bootstrap 5 • Tailwind CSS • Vue.js • React.js • Angular.js

Bootstrap 5

快速建構響應式網頁介面的CSS框架

 響應式設計 (RWD)

 豐富的組件庫

 快速原型開發

 管理後台首選

150+

預建組件

12

網格系統

HTML

```
<div class="container">
  <div class="row">
    <div class="col-md-6">
      <h1 class="display-4">標題</h1>
      <p class="lead">內容描述</p>
      <button class="btn btn-primary">
```

立即開始

```
</button>
    </div>
  </div>
</div>
```

適用場景



管理後台



快速原型



原子化CSS框架，極致的設計自由度

⚙️ 原子化設計理念

每個CSS類別只負責一個樣式屬性，如同原子般不可分割。透過組合這些「原子」類別，可以建構出任何想要的設計。

text-blue-500

藍色文字

p-4

內距 1rem

rounded-lg

大圓角

shadow-md

中等陰影

傳統 CSS 寫法

```
.custom-button{
  background-color: #3b82f6;
  color: white;
  padding: 0.5rem 1rem;
  border-radius: 0.375rem;
  border: none;
  cursor: pointer;
}
```

Tailwind CSS 寫法

```
<button class="bg-blue-500 text-white px-4 py-2 rounded-md border-none
  cursor-pointer">
  按鈕文字
</button>
```

📈 開發效率提升

50%

減少CSS代碼

3x

開發速度

90%

設計一致性



漸進式JavaScript框架，學習曲線平緩

MVVM 架構模式



雙向數據綁定，自動同步數據與視圖



易學易用

漸進式學習



響應式

自動更新視圖



組件化

可重用組件



輕量級

小巧靈活

Vue.js 組件範例

```
<template>
  <div class="counter">
    <h2>計數器: {{ count }}</h2>
    <button @click="increment">+1</button>
    <button @click="decrement">-1</button>
  </div>
</template>

<script>
export default {
  data() {
    return { count: 0 }
  },
  methods: {
    increment() { this.count++ },
    decrement() { this.count-- }
  }
}
</script>
```

學習曲線比較



Vue.js 擁有最平緩的學習曲線，適合初學者入門



Meta支援的組件化前端框架

組件化架構



可重用組件，提高開發效率



組件化

可重用組件



虛擬DOM

高效能渲染

React 函數組件範例

```
import React, { useState } from 'react';

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div className="counter">
      <h2>計數器: {count}</h2>
      <button onClick={() => setCount(count + 1)}>
+1
      </button>
      <button onClick={() => setCount(count - 1)}>
-1
      </button>
    </div>
  );
}

export default Counter;
```

豐富的生態系統

Next.js

全端框架

Redux

狀態管理

React Router

路由管理

Material-UI

UI組件庫

Styled Components

CSS-in-JS

React Native

移動端開發



Google支援的全功能企業級框架

全功能架構

應用層 (Components)

組件與模板

服務層 (Services)

業務邏輯與數據

路由層 (Router)

頁面導航

HTTP層 (HttpClient)

API通訊

完整的開發解決方案 · 開箱即用



全功能

完整解決方案



TypeScript

類型安全

Angular 組件範例

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-counter',
  template: `
    <div class="counter">
      <h2>計數器: {{count}}</h2>
      <button (click)="increment()">+1</button>
      <button (click)="decrement()">-1</button>
    </div>
  `
})
export class CounterComponent {
  count: number = 0;

  increment(): void {
    this.count++;
  }

  decrement(): void {
    this.count--;
  }
}
```



內建工具與功能

Angular CLI

命令行工具

RxJS

響應式編程



工具比較與選擇建議

CSS 框架

B

Bootstrap 5

快速建構響應式介面



組件豐富

快速開發

管理後台

T

Tailwind CSS

原子化CSS · 高度客製化



客製化

輕量化

現代設計

JavaScript 框架

V

Vue.js

漸進式框架 · 易學易用



學習簡單

漸進式

R

React.js

組件化 · 生態豐富



社群活躍

大型應用

A

Angular.js

全功能企業級框架



企業級

完整解決方案



選擇建議

初學者入門

中小型專案

大型企業應用

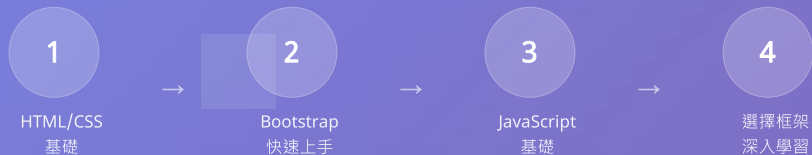


總結與未來趨勢

✓ 關鍵要點總結

- 選擇工具要考慮專案規模、團隊經驗和長期維護
- CSS框架：Bootstrap適合快速開發，Tailwind適合客製化
- JS框架：Vue易學，React生態豐富，Angular企業級
- 沒有最好的工具，只有最適合的工具

🧭 建議學習路徑



📈 未來發展趨勢



移動優先

響應式設計成為標準



組件化

可重用組件庫普及



性能優化

更快的載入與渲染



AI輔助

智能化開發工具

前端技術日新月異，**持續學習**是關鍵。
選擇適合的工具，**專注解決問題**，
而不是追逐最新的技術潮流。