Vue 3 面試重點技術指南

MVVM 架構

MVVM 是 Vue 的核心架構模式，面試重點：

- **Model(數據層)**：
  - 代表原始數據和業務邏輯
  - 通常是服務器返回的 JSON 數據或本地定義的數據結構

- **View(視圖層)**：
  - 使用者看到的界面
  - 以宣告式語法呈現數據
  - 包含模板語法如 `{{ }}`, `v-if`, `v-for` 等

- **ViewModel(視圖模型層)**：
  - Vue 實例作為中間層
  - 實現數據雙向綁定
  - 處理數據和視圖的同步更新

v-if vs v-for 優先級

重點掌握：

1. Vue 3 中 v-if 優先級高於 v-for（Vue 2 相反）
2. 避免同時使用，會影響性能且可能導致錯誤
3. 最佳實踐：
   - 使用計算屬性先過濾數據
   - 使用 template 標籤包裝

v-if vs v-show 選擇要點：

- v-if：適合條件較少改變的場景，完全銷毀和重建
- v-show：適合頻繁切換的場景，只改變 display 屬性

組件通信

重要的通信方式：

1. **Props Down**：
    - 父傳子的標準方式
    - 單向數據流，子組件不能直接修改
2. **Emit Up**：
    - 子組件觸發父組件事件
    - 可傳遞數據給父組件處理
3. **v-model**：
    - 實現雙向綁定
    - 底層是 props + emit 的語法糖

響應式數據

關鍵概念：

1. **ref**：
    - 適用於基本類型
    - 需要 .value 訪問
    - 可以解構保持響應性
2. **reactive**：
    - 用於對象類型
    - 直接訪問和修改
    - 解構會失去響應性

computed vs watch vs method

面試重點對比：

1. **computed**：
    - 有緩存機制
    - 依賴項變化才重新計算
    - 適合：數據計算且多次使用

2. **watch**：
    - 監聽數據變化
    - 可執行異步操作
    - 適合：數據變化後的後續操作
3. **method**：
    - 每次調用都執行
    - 可傳參數
    - 適合：事件處理和一次性計算

生命週期鉤子

重要的生命週期及使用場景：

1. **onMounted**：
    - DOM 已創建完成
    - 適合初始化和 API 調用
    - 可以訪問 ref 綁定的元素
2. **onUpdated**：
    - 數據更新後觸發
    - 注意避免無限循環更新
    - 適合需要訪問更新後 DOM 的操作
3. **onUnmounted**：
    - 組件銷毀前調用
    - 清理定時器、事件監聽等
    - 防止內存洩漏

效能優化技巧

1. **合理使用 v-show 和 v-if**：
    - 頻繁切換用 v-show
    - 條件穩定用 v-if

2. **列表渲染優化**：
    - 使用唯一 key
    - 避免不必要的組件渲染
3. **計算屬性緩存**：
    - 利用 computed 緩存計算結果
    - 避免模板中的複雜運算
4. **異步組件**：
    - 路由組件懶加載
    - 大型組件異步加載

Composition API 最佳實踐

1. **組合式函數(Composables)**：
    - 抽取可重用的邏輯
    - 保持單一職責原則
    - 明確的命名約定 (use 前綴)
2. **生命週期管理**：
    - 集中管理副作用
    - 清理註冊的事件和定時器
    - 避免內存洩漏
3. **狀態管理**：
    - 合理使用 ref 和 reactive
    - 提取共用狀態到 composables
    - 考慮使用 Pinia 進行全局狀態管理