

Tổng hợp lý thuyết về var, let, const trong JavaScript

Ngày 12 tháng 6 năm 2025

1 Scope (Phạm vi)

Scope là phạm vi mà biến có thể được truy cập. Trong JavaScript có 3 loại:

- **Global Scope:** Biến được khai báo bên ngoài hàm, truy cập được từ mọi nơi.
- **Function Scope:** Biến được khai báo bên trong một hàm, chỉ dùng trong hàm đó.
- **Block Scope:** Biến được khai báo trong cặp dấu ngoặc nhọn { }, chỉ truy cập được trong khối đó.
- **Ví dụ về code:** Hãy xem ví dụ trong file scope.js

2 Hoisting

- **var:** Được hoisting và khởi tạo với giá trị mặc định là `undefined`. Có thể truy cập trước khi khai báo.
- **let, const:** Cũng được hoisting nhưng không được khởi tạo. Nếu truy cập trước khi khai báo sẽ gây lỗi (*Temporal Dead Zone*).
- **Ví dụ về code:** Hãy xem ví dụ trong từng file var,let,const

3 Khai báo biến

- **var**: Có thể khai báo lại và gán lại. Có phạm vi toàn cục hoặc hàm.
- **let**: Không thể khai báo lại trong cùng một phạm vi, nhưng có thể gán lại. Có phạm vi khối.
- **const**: Không thể khai báo lại và không thể gán lại. Có phạm vi khối.
- **Ví dụ về code**: Hãy xem ví dụ trong file `scope.js`

4 Tái khai báo

- **var**: Cho phép tái khai báo trong cùng một phạm vi.
- **let, const**: Không cho phép tái khai báo trong cùng một phạm vi.
- **Ví dụ về code**: Hãy xem ví dụ trong file `scope.js`

5 Gán lại giá trị

- **var**: Có thể gán lại.
- **let**: Có thể gán lại.
- **const**: Không thể gán lại.
- **Ví dụ về code**: Hãy xem ví dụ trong file `scope.js`