**BIẾN và Kiểu dữ liệu:**

**BIẾN:**

* **Var:** phạm vi function, có hoisting, dễ lỗi
* **Let:** block scope, KHÔNG bị hoisting như var
* **Const:** giống let, nhưng không cho gán lại giá trị.

Hoisting là cơ chế mặc định của JavaScript để di chuyển tất cả các biến và hàm khi khai báo lên đầu scope trước khi chúng được thực thi.

**Kiểu dữ liệu cơ bản (Primitive)**:

* string, number, boolean, null, undefined, symbol, bigint

**Kiểu dữ liệu phức tạp (non-primitive)**:

* object, array, function

| **Kiểu biến** | **Gán lại** | **Phạm vi (Scope)** | **Ghi chú** |
| --- | --- | --- | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| var | ✅ | function | bị hoisting, hạn chế dùng |

|  |  |  |  |
| --- | --- | --- | --- |
| let | ✅ | block | dùng phổ biến hiện nay |

|  |  |  |  |
| --- | --- | --- | --- |
| const | ❌ | block | bắt buộc gán ngay khi khai báo |

**Câu hỏi ôn luyện (có đáp án)**

1. let và const khác gì nhau?  
   → let cho phép gán lại, const thì không.
2. Kiểu dữ liệu nào sau đây là *primitive*?  
   a. Object  
   b. Array  
   c. String ✅  
   d. Function
3. var x = 10; có thể bị hoisting không?  
   → **Có**, nhưng let và const thì không.

**Mini Quiz kiểm tra nhanh (Tự làm, rồi kiểm tra đáp án)**

A screen shot of a computer

AI-generated content may be incorrect.

<details> <summary>🎯 Xem đáp án</summary>

**Câu 1:** undefined → vì var bị hoisting.  
**Câu 2:** [1, 2, 3, 4] → const chỉ khóa *tham chiếu*, không khóa nội dung.  
**Câu 3:** 21 → let cho phép gán lại.

</details>

**Câu hỏi phỏng vấn thực tế liên quan**

**Q:** Sự khác nhau giữa var, let, const là gì?  
**A:** var có phạm vi function, bị hoisting. let và const có block scope. let gán lại được, const thì không.

**Bài tập nhỏ (Mini Project #1)**

Viết một đoạn JS để:

* Khai báo tên, tuổi, điểm trung bình.
* Lưu thành một object tên student.
* In ra câu: "Tên tôi là Tuan, năm nay tôi 22 tuổi, điểm TB là 9".