**Bài soạn Biến và kiểu dữ liệu trong JS - *Đỗ Văn Hoàng CGHN C0624G1***

|  |  |
| --- | --- |
| Từ khóa | Ghi chú |
| Biến?  Cú pháp khai báo biến?  Các cách khai báo biến?  Sự khác nhau giữa var, let, const?  Quy tắc đặt tên biến?  Kiểu dữ liệu?  Các kiểu dữ liệu?  Toán từ trong JS?  Độ ưu tiên của các toán từ?  == và === khác nhau như nào? | 1. Biến là gì?  * Biến là 1 tên gọi được gắn cho 1 vùng nhớ chứa dữ liệu. * Dữ liện được lưu trữ trong vùng nhớ của biến dữ liệu được gọi là giá trị (value). * Có thể truy nhập, gán, hay thay đổi giá trị của biển. * Khi gán 1 giá trị mới thì giá trị cũ sẽ bị ghi đè lên. * Cần phải khai báo biến trước khi sử dụng.  1. Cú pháp khai báo biến?  * Khai báo biến bằng var : * var variableName; * var anotherVariable = 10; * Khai báo biến bằng let : * let variableName; * let anotherVariable = "Hello"; * Khai báo biến bằng const : * const PI = 3.14; * const WEBSITE\_NAME = "My Website"; * Khai báo biến không cần khởi tạo: * var a; * let b; * const c; * Khai báo và gán giá trị ngay khi khai báo: * var x = 5; * let y = "JavaScript"; * const z = true; * Khai báo nhiều biến cùng một lúc: * var a, b, c; * let x = 1, y = 2, z = 3; * const PI = 3.14;  1. Có các cách khai báo biến nào? Phân biệt sự khác nhau giữa các cách?  * var : * Biến được khai báo bằng từ khóa “var”. * Có phạm vi là function-scoped, nghĩa là chỉ có hiệu lực trong function mà biến được khai báo. * Có thể gán giá trị mà không cần khởi tạo trước. * let : * Được sử dụng để khai báo biến có phạm vi block-scoped, chỉ tồn tại trong khối đó. * Không thể được khai báo cùng tên trong cùng một phạm vi. * Cần phải khởi tạo giá trị trước khi sử dụng. * const : * Tương tự như “let”, nhưng giá trị của biến không thể thay đổi sau khi đã được gán. * Cần phải gán giá trị ngay khi khai báo. * Cũng có phạm vi block-scoped. * Sự khác nhau giữa các cách khai báo :  |  |  |  | | --- | --- | --- | | var | let | const | | - Có phạm vi function-scoped.  - Có thể khai báo mà không cần gán giá trị.  - Có thể khai báo lại trong cùng một phạm vi. | - Có phạm vi block-scoped.  - Không thể khai báo cùng tên trong cùng một phạm vi.  - Cần phải gán giá trị trước khi sử dụng. | - Có phạm vi block-scoped.  - Giá trị không thay đổi sau khi gán.  - Cần phải gán giá trị ngay khi khai báo. |  1. Một số quy tắc đặt tên biến.  * Tên biến phải bắt đầu bằng một ký tự alphabet (a-zA-z\_). * Theo sau ký tự đầu có thể là các ký tự chữ, số … * Nên tránh đặt tên biến trùng tên các từ khoá. * Tên biến nên mô tả được ý nghĩa của nó. * Tránh dùng các ký tự gây lầm lẫn. * Tên biến có phân biệt chữ hoa và chữ thường. * Nên áp dụng các quy ước đặt tên biến chuẩn khi lập trình.  1. Kiểu dữ liệu là gì?  * Kiểu dữ liệu (Data type) cung cấp 1 bộ các giá trị mà từ đó 1 biểu thức (như biến, hàm, …) có thể lấy giá trị của nó. * Trong JS khi khai báo biến và gán cho biến 1 giá trị đồng nghĩa xác định kiểu dữ liệu cho biến đó.  1. Các kiểu dữ liệu trong JS?  * Chuỗi (string) : bao gồm các ký tự liên tiếp nhau, khai báo bằng dấu “ hoặc ‘. * Số (number) : dùng số nguyên hoặc số thập phân. * Boolean : true hoặc false. * …  1. Các toán tử trong JS :  * Toán tử toán học. * Toán tử gán. * Toán tử cộng chuỗi. * Toán tử so sánh. * Toán tử logic. * Toán tử typeof. * Độ ưu tiên trong các toán tử: * Trong một biểu thức có nhiều phép toán thì chúng sẽ lần lượt được đánh giá dựa vào độ ưu tiên. * Có thể sử dụng dấu ngoặc “()” để thay đổi độ ưu tiên của các toán tử. * Các toán tử có cùng độ ưu tiên thì sẽ thực hiện từ trái sang phải.      1. Phân biệt == và ===?  |  |  | | --- | --- | | == | === | | So sánh bằng giữa 2 giá trị mà không cần cùng kiểu dữ liệu. | So sánh bằng giữa 2 giá trị mà cần phải cùng kiểu dữ liệu. | |
|  | Kết luận |