**SOẠN BÀI BUỔI 7**

1. Biến là gì ?

- Biến là một tên gọi được gắn cho một vùng nhớ chứa dữ liệu.

2. Cú pháp khai báo biến ?

- Từ khóa var được dùng để khai báo biến

- x là tên biến

- Dấu (=) được dùng để gán giá trị cho biến

var variableName;

variableName = value

- VD: var x;

x = 6;

3. Các cách khai báo biến nào? Phân biệt sự khác nhau giữa các cách?

- Sử dụng var variableName = “”

+ Từ khóa var dùng để khai báo biến trên toàn cục.

+ Nếu bạn sử dụng từ khóa này để khai báo biến thì biến đó có thể truy cập toàn cục và cũng thay đổi được.

+ Cách này tuy đã lỗi thời nhưng vẫn có thể dùng tốt nếu số lượng dòng code ngắn. Ngược lại, bạn sẽ rất dễ bị nhầm lẫn.

- Sử dụng let variableName = “”

+ Từ khóa let dùng để khai báo biến cục bộ.

+ Nếu bạn sử dụng từ khóa này để khai báo biến thì biến đó có thể truy cập trong phạm vi cục bộ và cũng thay đổi được.

+ Cách khai báo biến này tốt kể cả khi số lượng dòng code trở nên nhiều hơn.

- Sử dụng const variableName = “”

+ Từ khóa const được sử dụng để khai báo biến cục bộ.

+ Nếu bạn dùng từ khóa này để khai báo biến thì biến chỉ có thể truy cập được trong khối đó, tương tự như biến được định nghĩa với let.

+ Và sự khác biệt giữa let và const là: bạn không thể gán lại giá trị cho các biến được khai báo với từ khóa const.

+ Vì vậy, bạn nên gán giá trị trong khi khai báo biến với const.

4. Nêu 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 …

- Tránh đặt tên biến trùng các từ khóa

- 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

5. Kiểu dữ liệu là gì ?

- Một kiểu dữ liệu cung cấp một bộ giá trị mà từ đó một biểu thức (như biến, hàm …) có thể lấy giá trị của nó

- Trong JS khi khai báo biến và gán cho biến một giá trị đồng nghĩa xác định kiểu dữ liệu cho biến đó

6. Các kiểu dữ liệu trong JS ?

- Number: bất kỳ loại số nào số nguyên hoặc số thực.

- String: Chuỗi có thể có một hoặc nhiều ký tự, không có kiểu ký tự đơn nào khác nhau.

- Giá trị boolean: true/false.

- Null: không có giá trị nào thỏa mãn.

- Undefined: giá trị chưa được gán hoặc giá trị không xác định.

- Object: Cấu trúc dữ liệu phức tạp hơn, thể hiện một đối tượng và các thuộc tính có thể truy cập đến.

7. Các toán tử trong JS ? Độ ưu tiên các toán tử ?

- 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 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

8. Phân biệt giữa == và === ?

- Toán tử **==** so sánh trừu tượng (abstract equality), tức là nó thực hiện các chuyển đổi loại cần thiết trước khi so sánh đẳng thức. Cú pháp so sánh: **a == b**.

- Toán tử **===** so sánh cân bằng nghiêm ngặt (strict equality), nghĩa là nó sẽ không thực hiện chuyển đổi loại. Do đó nếu hai giá trị không cùng loại, thì khi so sánh, kết quả sẽ trả về **false.**Cú pháp so sánh: **a === b**.