Câu 1:

Khái niệm vòng lặp: Vòng lặp là một cấu trúc trong lập trình cho phép lặp đi lặp lại một khối lệnh nhiều lần cho đến khi một điều kiện nào đó không còn đúng.

Tại sao cần sử dụng vòng lặp: Vòng lặp giúp tiết kiệm thời gian, giảm lặp code, và xử lý các tác vụ lặp đi lặp lại một cách tự động thay vì viết thủ công nhiều lần.

Ví dụ:

for (let i = 1; i <= 5; i++) {  
 console.log("Số thứ tự: " + i);  
}

Câu 2:

Các loại vòng lặp trong JavaScript: for, while, do..while, for..in, for..of.

Biểu thức và luồng thực thi của for: for (khởi tạo; điều kiện; cập nhật) { khối lệnh }

Thứ tự: Thực hiện khởi tạo → Kiểm tra điều kiện → Nếu đúng thì chạy khối lệnh → Sau đó cập nhật → Quay lại kiểm tra điều kiện → Lặp lại.

Ví dụ:

for (let i = 0; i < 5; i++) {  
 console.log("Giá trị i: " + i);  
}

Câu 3:

Cách sử dụng while: Lặp lại khi điều kiện đúng.

Ví dụ:

let i = 1;  
while (i <= 5) {  
 console.log("i = " + i);  
 i++;  
}

Cách sử dụng do..while: Thực hiện ít nhất một lần, sau đó mới kiểm tra điều kiện.

Ví dụ:

let j = 1;  
do {  
 console.log("j = " + j);  
 j++;  
} while (j <= 5);

Câu 4:

Giống nhau: Đều dùng để lặp đi lặp lại một khối lệnh khi điều kiện thỏa mãn.

Khác nhau:

- for: Biểu thức gọn gàng, dễ theo dõi khi biết trước số lần lặp.

- while: Thích hợp khi không biết trước số lần lặp, chỉ dựa vào điều kiện.

- do..while: Luôn thực hiện ít nhất một lần trước khi kiểm tra điều kiện.

Có thể thay thế nhau không: Có thể thay thế trong hầu hết các trường hợp, nhưng lựa chọn đúng vòng lặp giúp code dễ hiểu và hiệu quả hơn.

Câu 5:

break: Dùng để thoát hẳn khỏi vòng lặp khi điều kiện nào đó xảy ra.

Ví dụ:

for (let i = 1; i <= 10; i++) {  
 if (i === 5) break;  
 console.log(i);  
}  
// Kết quả: 1, 2, 3, 4

continue: Dùng để bỏ qua lần lặp hiện tại và chuyển sang vòng lặp tiếp theo.

Ví dụ:

for (let i = 1; i <= 5; i++) {  
 if (i === 3) continue;  
 console.log(i);  
}  
// Kết quả: 1, 2, 4, 5