

Loop condition in JS

Date: 25/12/2024	
Topic	Subject
Vòng lặp trong js	
<div>Essential Questions:</div> <div>1. Khái niệm vòng lặp? Tại sao cần phải sử dụng vòng lặp trong lập trình. Cho 1 ví dụ</div> <div>2. Các loại vòng lặp trong JS? Các biểu thức và luồng thực thi của for? Cho ví dụ về vòng lặp for?</div> <div>3. Các sử dụng while và do while. Cho ví dụ cho từng vòng lặp?</div> <div>4. So sánh giống và khác nhau giữa for, while và do..while? Liệu rằng các vòng lặp trên có thể thay thế cho nhau được không</div> <div>5. Phân biệt break và continue? Cho ví dụ trường hợp dùng break, trường hợp dùng continue</div>	
Main Ideas	Notes
	<div>Cornell Note: Vòng Lặp và Điều Khiển Luồng trong JavaScript</div> <div>Main Topic: Vòng Lặp (Loop)</div> <div>Notes</div> <div>1. Khái niệm vòng lặp:</div> <div><ul style="list-style-type: none">Định nghĩa: Vòng lặp là một cấu trúc trong lập trình cho phép thực thi một đoạn mã nhiều lần dựa trên điều kiện nhất định.Lợi ích: Giảm lặp mã, tăng hiệu quả và dễ bảo trì.Ví dụ: <pre>for (let i = 0; i < 5; i++) { console.log(i); }</pre></div> <div>2. Các loại vòng lặp trong JS:</div> <div><ul style="list-style-type: none">For: Thực thi một đoạn mã với số lần xác định.<ul style="list-style-type: none">Biểu thức: <pre>for (initialization; condition; increment/decrement) { }</pre><ul style="list-style-type: none">Ví dụ: <pre>for (let i = 1; i <= 5; i++) { console.log(i); // In từ 1 đến 5 }</pre>While: Thực thi khi điều kiện đúng, số lần không cố định.<ul style="list-style-type: none">Cú pháp: <pre>while (condition) { }</pre><ul style="list-style-type: none">Ví dụ: <pre>let i = 1; while (i <= 5) { console.log(i); i++; }</pre>Do-While: Luôn thực thi ít nhất một lần, sau đó kiểm tra điều kiện.<ul style="list-style-type: none">Cú pháp: <pre>do { } while (condition);</pre><ul style="list-style-type: none">Ví dụ: <pre>let i = 1; do { console.log(i); i++; } while (i <= 5);</pre></div> <div>3. So sánh for, while, do-while:</div> <div><ul style="list-style-type: none">Giống nhau:<ul style="list-style-type: none">Đều dùng để lặp một đoạn mã.Có thể thay thế nhau bằng cách thay đổi logic.Khác nhau:<ul style="list-style-type: none">For: Biểu thức rõ ràng, phù hợp khi biết trước số lần lặp.While: Phù hợp khi số lần lặp không xác định, kiểm tra điều kiện trước khi thực thi.Do-While: Thực thi ít nhất một lần trước khi kiểm tra điều kiện.</div> <div>4. Liệu các vòng lặp có thể thay thế nhau được không?</div> <div><ul style="list-style-type: none">Có thể thay thế nhau nếu viết lại điều kiện và logic tương ứng.Tuy nhiên, chọn đúng loại vòng lặp giúp mã dễ đọc và hiệu quả hơn.</div> <div>5. Phân biệt break và continue:</div> <div><ul style="list-style-type: none">Break: Dừng hoàn toàn vòng lặp.<ul style="list-style-type: none">Ví dụ: Dừng vòng lặp khi gặp số chẵn. <pre>for (let i = 1; i <= 10; i++) { if (i % 2 === 0) break; console.log(i); }</pre>Continue: Bỏ qua lần lặp hiện tại, tiếp tục vòng lặp.<ul style="list-style-type: none">Ví dụ: Bỏ qua số chẵn, chỉ in số lẻ. <pre>for (let i = 1; i <= 10; i++) { if (i % 2 === 0) continue; console.log(i); }</pre></div>
Summary	
Vòng lặp là công cụ mạnh mẽ trong lập trình để lặp lại các tác vụ. Các loại vòng lặp như for, while, và do-while có đặc điểm riêng và có thể thay thế nhau tùy trường hợp. Lệnh break và continue giúp điều khiển luồng vòng lặp hiệu quả, tùy theo yêu cầu cụ thể.	

