

JavaScript Loops Review

Working with Loops

- **for Loop:** This type of loop is used to repeat a block of code a certain number of times. This loop is broken up into three parts: the initialization statement, the condition, and the increment/decrement statement. The initialization statement is executed before the loop starts. It is typically used to initialize a counter variable. The condition is evaluated before each iteration of the loop. An iteration is a single pass through the loop. If the condition is true, the code block inside the loop is executed. If the condition is false, the loop stops and you move on to the next block of code. The increment/decrement statement is executed after each iteration of the loop. It is typically used to increment or decrement the counter variable.

Example Code

```
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}
```

- **for...of Loop:** This type of loop is used when you need to loop over values from an iterable. Examples of iterables are arrays and strings.

Example Code

```
const numbers = [1, 2, 3, 4, 5];  
  
for (const num of numbers) {  
  console.log(num);  
}
```

- **for...in Loop:** This type of loop is best used when you need to loop over the properties of an object. This loop will iterate over all enumerable properties of an object, including inherited properties and non-numeric properties.

Example Code

```
const fruit = {
```

```
name: 'apple',  
color: 'red',  
price: 0.99  
};
```

```
for (const prop in fruit) {  
  console.log(fruit[prop]);  
}
```

- **while Loop:** This type of loop will run a block of code as long as the condition is true.

Example Code

```
let i = 5;
```

```
while (i > 0) {  
  console.log(i);  
  i--;  
}
```

- **do...while Loop:** This type of loop will execute the block of code at least once before checking the condition.

Example Code

```
let userInput;
```

```
do {  
  userInput = prompt("Please enter a number between 1 and 10");  
} while (Number(userInput) < 1 || Number(userInput) > 10);
```

```
alert("You entered a valid number!");
```

break and continue Statements

- **Definition:** A break statement is used to exit a loop early, while a continue statement is used to skip the current iteration of a loop and move to the next one.

Example Code

// Example of break statement

```
for (let i = 0; i < 10; i++) {  
  if (i === 5) {  
    break;  
  }  
  console.log(i);  
}
```

// Output: 0, 1, 2, 3, and 4

// Example of continue statement

```
for (let i = 0; i < 10; i++) {  
  if (i === 5) {  
    continue;  
  }  
  console.log(i);  
}
```

// Output: 0, 1, 2, 3, 4, 6, 7, 8, and 9