Day 5:-

Today I learned about Loops and Arrays.

1. Loops:

In JavaScript, a loop is a control flow statement that allows a block of code to be executed repeatedly a certain number of times or until a certain condition is met. Common types of loops in JavaScript include for loops, while loops, and for-each loops. Each type has its own syntax and use cases. Here are the types of loops in JavaScript along with their syntax and examples:

1. for loop: This type of loop is used to iterate over a sequence of elements (such as an array or string) a specific number of times. Syntax:

Syntax:

for (initialization; condition; increment/decrement) {

// code to be executed

}

Example:

for(var i = 0; i < 10; i++) {

console.log(i);

}

1. for...in loop: This type of loop is used to iterate over the properties of an object. It is similar to the for loop, but it loops over the properties of an object instead of an array. Syntax:

syntax:

for (variable in object) {

// code to be executed

}

Example:

var person = {name: "John", age: 30, city: "New York"};

for (var x in person) {

console.log(x);

}

1. for...of loop: This type of loop is similar to the for...in loop, but it is used to iterate over the values of an iterable object (such as an array, set or map) instead of the properties of an object. Syntax:

for (variable of iterable) {

// code to be executed

}

Example:

var cars = ["BMW", "Volvo", "Saab", "Ford"];

for (var i of cars) {

console.log(i);

}

1. while loop: This type of loop is used to execute a block of code repeatedly as long as a certain condition is true. Syntax:

syntax:

while (condition) {

// code to be executed

}

Example:

var i = 0;

while (i < 10) {

console.log(i);

i++;

}

1. do...while loop: This type of loop is similar to the while loop, but the block of code is executed at least once before the condition is checked. Syntax:

syntax:

do {

// code to be executed

} while (condition);

Example:

var i = 0;

do {

console.log(i);

i++;

} while (i < 10);

Although break and continue are not types of loops, but rather control flow statements that can be used in conjunction with loops. I have mentioned them as they are used in loop.

1. break and continue statement:

break and continue are control flow statements in JavaScript that are used to control the flow of a loop.

break statement is used to exit the loop early, before the loop has finished executing all of its iterations. When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.

Copy code

for (var i = 0; i < 10; i++) {

if (i === 5) {

break;

}

console.log(i);

}

continue statement is used to skip one iteration of the loop. When a continue statement is encountered inside a loop, the current iteration of the loop is skipped and the program control resumes with the next iteration of the loop.

Copy code

for (var i = 0; i < 10; i++) {

if (i % 2 === 0) {

continue;

}

console.log(i);

}

It's worth noting that these statements can be used in all types of loops (for, for...in, for...of, while, do...while) in javascript.