CodeWithHarry

HTML CSS JS C C++ JAVA PYTHON PHP

For, While & Do while Loops in JavaScript | JavaScript Tutorial In Hindi #9

Overview Q&A Downloads Announcements

For, While & Do while Loops in JavaScript | JavaScript Tutorial In Hindi #9

There will be times when we want to repeat an action or run some code multiple times. Suppose we want to display the message 100 times. We could do this by simply creating a function display_Message() and calling it 100 times using copy and paste. No doubt, this will work but, we should not do something like this. Duplicating code is never a good idea. The versatility of the computer lies in its ability to perform the set of instructions repeatedly. The generic solution for repeating code with control is provided in the form of loops. There are four different kinds of loops in JavaScript that we can be used to repeat some chunk of code:

- **for**
- **for_in**
- **forEach**
- **while**
- **do_while**

The for Loop:-

The for loop is used when we want to execute the set of statements for a certain number of times. This loop will repeat a block of code as long as a certain condition is met. Its syntax is:

```
for(initialization; condition; increment) {
    // Code to be executed
  }
```

- ***initialization***:- It is used to initialize the counter variables, for example, let i=0.
- ***Condition***:- The condition is evaluated at the beginning of each iteration. If it evaluates to true, the loop statements execute, and If it evaluates to false, the

execution of the loop ends, i.e., i<100.

- ***Increment***:- it updates the loop counter with an incremented value value each time the loop runs for example i++;

Following is the example of for loop:

```
for(var i=1; i<=5; i++) {
console.log(i);
}
```

The for... in Loop:-

It is a special type of loop, used when we want to iterates over the properties of an object or the elements of an array. The syntax for using the for-in loop is:

```
for(variable in object) {
  // Code to be executed
 }
```

The loop counter, i.e., *variable* in the for-in loop, is a string, not a number. It contains the name of the current property or the index of the current array element. The following example will show how to loop through all the properties of a JavaScript object.

```
// An object with some properties
var person = {name: "Harry", language: "JavaScript", age: 20};
// Loop through all the properties in the object
for(var i in person) {
console.log( i + " = " + person[i]);
}
```

The forEach Loop:-

In JavaScript, the forEach is a type of loop that is used for Array method. With the help of forEach loop, we can execute a function on each item within an array. The function can only be used on Arrays, Sets, and Maps.

Example:-

```
const  alpha = ['a', 'b', 'c'];
for (let i = 0; i < alpha.length; i++) {
console.log(alpha[i]);
}
```

while loop:-

A while loop is used when we do not know how many times a certain block of code should execute. It evaluates the expression inside the parenthesis (). If the

expression evaluates to true, the code inside the while loop is executed. Every time the expression is re-evaluated, the process continues until the expression evaluates to false. When the expression evaluates to false, the loop stops.
The syntax of the while loop is:

```
while (expression) {
  // body of loop
}
```

For Example:-

```
let i = 1, n = 100;
while (i <= n) {
    console.log(i);
    i += 1;
}
```

do...while loop:-

In this loop, the body inside the do statement is executed first. Then the condition is evaluated. If the condition evaluates to true, the body of the loop inside the do statement is executed again. This process continues until the condition evaluates to false. Then the loop stops.
The syntax of do...while loop is:

```
do {
    // body of loop
} while(condition)
```

For Example:-

```
let i = 1;
let n =100;
do {
    console.log(i);
    i++;
} while(i <= n);
```

Note: do...while loop is similar to the while loop. The only difference is that in do...while loop, the body of loop is executed at least once.

Breaking a Loop:-

Sometimes, we may want to end our loop before it reaches completion. By using break statement, we can accomplish this task. Here is an example:

```
for (let i = 0; i < 100; i++) {
 console.log(i);
    if (i == 90) {
        break;
    }
}
```

When the value of i equals **90**, the break keyword stops the loop.

Skipping an Iteration:-

When we want the loop to skip its current iteration and move on to the next one we use “continue” keyword. Here is an example:

```
let i = 10;
for (let x = 1; x <=i; x++) {
    if (x == 5) {
        continue;
    }}
```

As mentioned above, continue will stop the loop when x becomes equal to 5 and move on to the next iteration. In handling errors, continue keyword is also very useful when we want the loop to move on to the next item.

Code index.html as described/written in the video

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Tutorial on Js</title>
</head>
<body>
    <h1>This is Js tutorial by Harry</h1>
</body>

<!-- <script src="js/tut2.js"></script> -->
<!-- <script src="js/tut3.js"></script> -->
```

```
<!-- <script src="js/tut4.js"></script> -->
<!-- <script src="js/tut5.js"></script> -->
<!-- <script src="js/tut6.js"></script> -->
<!-- <script src="js/tut7.js"></script> -->
<!-- <script src="js/tut8.js"></script> -->
<!-- <script src="js/tut9.js"></script> -->
<!-- <script src="js/tut10.js"></script> -->
<script src="js/tut11.js"></script>
</html>
```

Js code as described/written in the video

```
  console.log("We are at tutorial 9");
// console.log(1)
// console.log(2)
// console.log(3)

// General Loops: for, while, do-while
// let a =34;
// a +=1;
// a++;
// console.log(a);
// for(let i=0; i<100;i++){
//     console.log(i);
// }

// let j = 110;
// while (j < 10) {
//   console.log(j + 1);
//   j += 1;
// }

// let k = 0;
// do {
//     if(k===5){
//         k +=1;
//         continue;
//     }
//   console.log(k + 1);
//   k +=1;
// } while (k < 10);

let arr = [2,5,6,4,2,3];

// for (let i = 0; i < arr.length; i++) {
//     const element = arr[i];
//     console.log(element)
// }

arr.forEach(function(element, index, array){
    console.log(element, index, array);
});

// let obj = {
//     name: "Rohan Das",
//     age: 78,
//     type: "Dangerous Programmer",
//     os: "Ubuntu"
// }
// for(let key in obj){
//     console.log(`The ${key} of object is ${obj[key]}`)
// }

console.log('done');
```

Previous Next

CodeWithHarry