codingninjas

Conditional and Looping statement in javascript

If Statement:

The if statement is used for conditional execution of code. It evaluates a condition and, if the condition is true, executes a block of code.

```
if (condition) {
  // Code to be executed if the condition is true
}
```

Example:

```
let num = 10;

if (num > 0) {
   console.log("The number is positive"); // Output: The number is positive
}
```

If-Else Statement:

The if-else statement extends the if statement by providing an alternative block of code to be executed if the condition is false.

```
  if (condition) {
  // Code to be executed if the condition is true
} else {
  // Code to be executed if the condition is false
}
```

Example

```
let num = -5;

if (num > 0) {
  console.log("The number is positive");
} else {
  console.log("The number is non-positive"); // Output: The number
is non-positive
}
```

If-Else-If Statement:

The if-else-if statement allows you to check multiple conditions and execute different blocks of code based on which condition is true.

```
if (condition1) {
 // Code to be executed if condition1 is true
} else if (condition2) {
 // Code to be executed if condition2 is true
} else {
 // Code to be executed if none of the conditions are true
}
```

Example

```
let num = 0;

if (num > 0) {
  console.log("The number is positive");
} else if (num < 0) {
  console.log("The number is negative");
} else {
  console.log("The number is zero"); // Output: The number is zero
}
```

Switch Statement:

The switch statement is used to perform different actions based on different conditions. It is often more concise than a series of if-else-if statements.

```
switch (expression) {
 case value1:
 // Code to be executed if expression equals value1
 break;
 case value2:
 // Code to be executed if expression equals value2
 break;
 // Additional cases as needed
 default:
 // Code to be executed if none of the cases match
}
```

Example:

```
switch (day) {
  case "Monday":
     console.log("It's the start of the week"); // Output: It's
the start of the week
    break;
  case "Friday":
    console.log("It's almost the weekend");
    break;
  default:
    console.log("It's a regular day");
}
```

For Loop:

The for loop is used when you know the number of iterations in advance.

```
for (initialization; condition; iteration) {
 // Code to be executed in each iteration
}
```

Example

```
  for (let i = 0; i < 5; i++) {
  console.log("Iteration " + (i + 1));
}
// Output:
// Iteration 1
// Iteration 2
// Iteration 3
// Iteration 4
// Iteration 5
```

While Loop:

The while loop is used when the number of iterations is not known in advance, and it continues iterating as long as the specified condition is true.

```
while (condition) {
 // Code to be executed as long as the condition is true
}
```

Example

```
let i = 0;

while (i < 5) {
  console.log("Iteration " + (i + 1));
  i++;
}
// Output:
// Iteration 1
// Iteration 2
// Iteration 3
// Iteration 4
// Iteration 5
```

Do-While Loop:

Similar to the while loop, but it guarantees that the code inside the loop will be executed at least once, as the condition is checked after the execution of the loop body.

```
do {
 // Code to be executed at least once
} while (condition);
```

Example

```
                  let i = 0;

do {
  console.log("Iteration " + (i + 1));
  i++;
} while (i < 4);
// Output:
// Iteration 1
// Iteration 2
// Iteration 3
// Iteration 4
```

Nested Loops:

You can use loops inside other loops to create nested structures for more complex iterations.

```
for (let i = 0; i < 5; i++) {
 for (let j = 0; j < 3; j++) {
 // Code to be executed for each combination of i and j
 }
}
```

Example

```
for (let i = 0; i < 3; i++) {
  for (let j = 0; j < 2; j++) {
    console.log(`i: ${i}, j: ${j}`);
  }
}
// Output:
// i: 0, j: 0
// i: 0, j: 1
// i: 1, j: 0
// i: 1, j: 1
// i: 2, j: 0
// i: 2, j: 1
```

References :

If statement : Click here to read more

Switch statement : click here to read more

Looping : Click here to read more