Google | طويق

JavaScript II

Conditionals and loops

أكاديمية طويق
TUWAIQ ACADEMY

وزارة الاتصالات وتقنية المعلومات
MINISTRY OF COMMUNICATIONS AND INFORMATION TECHNOLOGY

الاتحاد السعودي للأمن السيبراني والبرمجة والدرونز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

مهارات المستقبل

stc

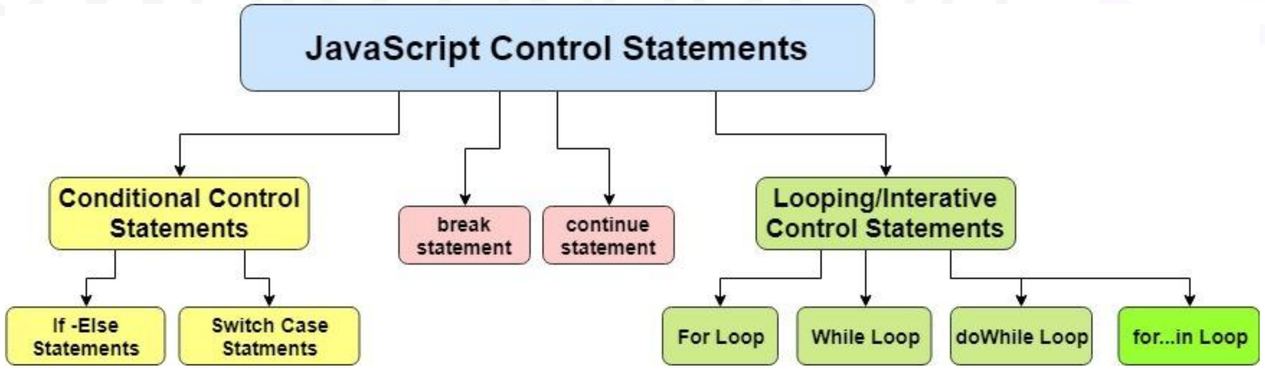
Objective

- Comparison Operators
- If, else and else if conditional
- Logical Operators
- Switch Conditional
- While loop, do while loop and for loop
- Infinite loops and nested loops

Control Structures in JavaScript

- The control structures allow the flow of your program to change within a unit of code or function.

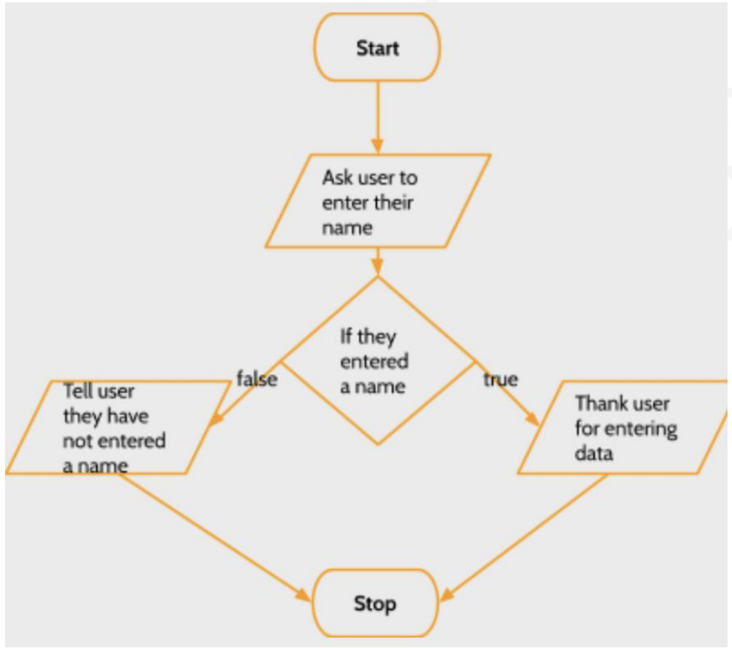
- These statements can determine whether given statements are executed, or they can repeat the execution of a block of code.



The if Conditional

- **If Statement:** is able to compare two or more variables or scenarios and perform a certain action based on the outcome of that comparison

- In JavaScript if statements have the following general syntax:




```
If (condition) {
        Indented statements;
}
```

Comparison Operators

Operator	Description	Example
>	greater than	Condition: 12 > 1 Result: True
<	less than	Condition: 12 < 1 Result: False
>=	greater than or equal to	Condition: 12 >= 1 Result: True
<=	less than or equal to	Condition: 12 <= 12 Result: True
==	equals	Condition: 12 == 1 Result: False
===	Equal value and equal type	Condition: 12 === twelve Result: False
!=	does not equal	Condition: 12 != 1 Result: True

ELSE and ELSE IF Statement

```
let num = 10;

if (num >12) {
    console.log("the variable num is greater than 12");
}
else if (num > 10) {
    console.log("the variable num is greater than 10");
}
else if (num < 5) {
    console.log("the variable num is less than 5");
}
else {
    console.log("the variable num is 10");
}
```

Logical Operation

- AND (&&):

```
let num = 12;
if (num >= 10 && num <= 15){
    console.log(num + " is a value between 10 and 15");
}
```

- OR (||):

```
let lotsOfMoney = false;
let receivedGift = false;
let loanApproved = true;

if (lotsOfMoney || receivedGift || loanApproved){
    console.log("Can purchase a car");
} else {
      console.log("Sorry! Can't afford a car");
}
```

- NOT (!)

The Switch Conditional

```
switch(expression)
{
	case 1:
		Do something;
		break;
	case 2:
		Do something else;
		break;
	default:
		Else do this;
		break;
}
```

Repetition or Looping Control Structures

- The while loop: will repeatedly execute a block of code for as long as the evaluation condition is met.

```
let i = 0;
while (i < 10) {
    i++;                    // shorthand for i = i + 1
    console.log(i);
}
```

Repetition or Looping Control Structures

❖ **The for loop:** very similar functionality to a while loop, but runs a predetermined number of times.

```
for (i = 1; i < 10; i++) {
        console.log(i);
}
```

If statement vs Conditional (Ternary) Operator

```
let age = 15;

if(age<18){

  console.log("Too young")

}

else{

  console.log("Old enough")

}
```

```
let age = 15;

let voteable = (age < 18) ? "Too young":"Old
enough";


console.log(voteable)
```

Break & Continue statement

- **Break statement** causes an immediate exit from the loop to the first statement after the loop body.

```
for (let i = 0; i < 10; i++) {
  if (i === 3) {
        break;
         }
  console.log("The number is " + i );
}
```

- **Continue statement** breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

```
for (let i = 0; i < 10; i++) {
  if (i === 3) {
        continue;
         }
  console.log("The number is " + i );
}
```

Nested loop

- A nested loop is simply a loop within a loop.

- Each time the outer loop is executed, the inner loop is executed right from the start. That is, all the iterations of the inner loop are executed with each iteration of the outer loop.

```
for (iterating_var in sequence) {
   for (iterating_var in sequence) {
      statements(s);
   }
   statements(s);
}
```

Nested for loop in another for loop

```
while (condition) {
   while (condition) {
      statement(s);
   }
   statement(s);
}
```

Nested while loop in another while loop

```
for (iterating_var in sequence) {
  while (condition) {
     statement(s);
  }
  statements(s);
}
```

Nested for loop in another while loop

Which Loop to Choose?

- Both the while loop and the for loop have 4
- **Common components:**
  - Initialization of control variable
  - Termination condition
  - Update control variable
  - Body to be repeated
- A while loop is generally used when we **don’t know how many times** to run through the loop
- If we **do know how many times** to run through the loop, we use the for loop

Resources

- JavaScript conditional statements and loops - Exercises, Practice, Solution

Summary

- Comparison Operators
- If, else and else if conditional
- Logical Operators
- Switch Conditional
- While loop, do while loop and for loop
- Infinite loops and nested loops