

Control Structures

Most of the programming languages use control structures to control the flow of a program. The control structures include decision-making and loops. Decision-making is done by applying different conditions in the program. If the conditions are true, the statements following the condition are executed. The values in a condition are compared by using the comparison operators. The loops are used to run a set of statements several times until a condition is met. If the condition is true, the loop is executed. If the condition becomes false, the loop is terminated and the control passes to the next statement that follows the loop block.

The control structures that are used in JavaScript are as follows:

The if statement: The if statement is a simple decision-making statement that is used to apply conditions in the program. If a condition is true, the statement in the if block is executed; otherwise, the statement following the if block is executed.

The syntax of the if statement is as follows:

```
if(condition)
{
    //statements
}
```

The following example will demonstrate the use of the if statement:

```
<script language="javascript">
function ifdemo()
{
    var a=prompt("Enter a number")
    var b= prompt("Enter a number")
    if(a>b)
    {
        document.write(a)
    }
}
</script>
```

The above code will execute only if the value of the variable a is greater than the value of the variable b, and the value of the variable a will be printed.

The if-else statement: The if-else statement is used to apply a condition in the program. If the condition is true, the statement in the if block is executed. If the condition is false, the statement in the else block is executed.

The syntax for using the if-else statement is as follows:

```
if(condition)
{
    //statements
}
else
{
```

```
    //statements
}
```

The following example will demonstrate the use of the if else statement:

```
<script language="javascript">
function ifelsedemo()
{
    var a=prompt("Enter a number")
    var b= prompt("Enter a number")
    if(a>b)
    {
        document.write(a)
    }
    else
    {
        document.write(b)
    }
}
</script>
```

In the above code, if the value of the variable a is greater than the value of the variable b, the value of the variable a will be printed. If the value of the variable b is greater than the value of the variable a, the value of the variable b will be printed.

The if-else if statement: The if-else if statement is used to check several conditions. If any of the condition is true, the statement of that block is executed; otherwise, the value in the else block is executed.

The syntax of using the if-else if statement is as follows:

```
if(condition)
{
    //statements
}
else if(condition)
{
    //statements
}
else if(condition)
{
    //statements
}
else
{
    //statements
}
```

The following example will demonstrate the use of the if-else if statement:

```
<script language="javascript">
function ifelseifdemo()
{
```

```

var a=parseFloat(prompt("Enter the marks in Math"))
var b=parseFloat(prompt("Enter the marks in Science"))
var c=parseFloat(prompt("Enter the marks in Geography"))
var d=parseFloat(prompt("Enter the marks in History"))
var e=parseFloat(prompt("Enter the marks in Commerce"))
var f=(a+b+c+d+e)*100/500
if(f<40)
{
    alert("Fail")
}
else if(f>=40 && f<50)
{
    alert("Fair")
}
else if(f>=50 && f<60)
{
    alert("Good")
}
else if(f>=60 && f<75)
{
    alert("Very Good")
}
else if(f>75)
{
    alert("Excellent")
}
}
</script>

```

In the above code, five conditions are given. If anyone of the condition is true, the statement following the block will be executed.

The conditional statement: The conditional statement can be used in the same way as the if else statement. The conditional statement is a single line statement that can be used in place of the if-else statement.

The syntax of using the conditional statement is as follows:

```
variablename=condition ? value1 : value2
```

If the condition is true, the statement following the question mark(?) will be executed. If the condition is false, the statement after the colon(:) is executed.

The following code will demonstrate the use of the conditional expression:

```

<script language="javascript">
function condexp()
{
    var a=prompt("Enter a number")
    var b=prompt("Enter a number")
    var c=a>b ? a : b
    document.write(c)
}

```

</script>

Loops: The loops are the statements that are used to run a set of statements several times until a condition is met. These statements can also be used to extract the elements of an array.

The loops that are used in JavaScript are as follows:

The for loop: The for loop is created by using the for keyword.

The syntax of using the for loop is as follows:

```
for(initialization; condition; increment/decrement)
{
    //statements
}
```

In the above syntax, the initialization refers to the value that starts the loop. The condition is the conditional statement that is used to control the loop. If the condition is true, the loop continues. If the condition is false, the loop is terminated and the control passes to the next statement that follows the loop block. The increment or decrement is used to run the loop in forward or backward direction.

The following code will demonstrate the use of the for loop:

```
<script language="javascript">
function fordemo()
{
    var a
    for(a=1; a<=10; a++)
    {
        document.write(a)
    }
}
</script>
```

The above code will print a series from 1 to 10.

The for in loop: The for in loop is the variation of the for loop. It is used to extract the properties of the object that are being used in the current browser window.

The syntax of using the for in loop is as follows:

```
for(counter in object)
{
    //statements
}
```

The following code will demonstrate the use of the for in loop

```
< script language="javascript">
function forindemo()
{
```

```

var obj=document
var objname="window"
var result=""
var i
for(i in obj)
{
    result+=objname+"."+i+"="+obj[i]+ "<br>"
}
document.write(result)
}
</script>

```

The while loop: The while loop is a statement that is used to run a statement or a set of statements several times until a condition is met. The while loop checks the condition and runs the loop until the condition is true. If the condition becomes false, the loop is terminated and the control goes to the next statement that follows.

The syntax of the while loop is as follows:

```

while(condition)
{
    //statements
}

```

The following example will demonstrate the use of the while loop:

```

<script language="javascript">
function whiledemo()
{
    var num=0
    while(num<=10)
    {
        document.write(num)
    }
}
</script>

```

The above statement will print a series from 1 to 10.

The do-while loop: The do-while loop is a statement that is used to run a statement or a set of statements several times until a condition is met. The do-while loop starts before checking a condition. After its first execution, it checks the condition and runs the loop until the condition is true. If the condition becomes false, the loop is terminated and the control goes to the next statement that follows.

The syntax of the do-while loop is as follows:

```

do
{
    //statements
} while(condition)

```

The following example will demonstrate the use of the do-while loop:

```

<script language="javascript">
function dowhiledemo()
{
    var num=0
    do
    {
        document.write(num)
    }while(num<=10)
}
</script>

```

The above statement will print the series from 1 to 10.

The break statement: The break statement is used to terminate a loop according to the given condition inside the loop.

The syntax of using the break statement is as follows:

```

for(initialization; condition; increment/decrement)
{
    //statements
    if(condition)
    {
        break
    }
}

```

The following code will demonstrate the use of the break statement:

```

<script language="javascript">
function breakdemo()
{
    var a
    for(a=1; a<=10; a++)
    {
        document.write(a)
        if(a==5)
        {
            break
        }
    }
}
</script>

```

The above code will print a series from 1 to 5.

The continue statement: The continue statement breaks the loop and starts the loop again with the next value.

The syntax of using the continue statement is as follows:

```

for(initialization; condition; increment/decrement)

```

```

{
    //statements
    if(condition)
    {
        continue
    }
}

```

The following example will demonstrate the use of the continue statement:

```

<script language="javascript">
function cont()
{
    var a
    for(a=0; a<10; a++)
    {
        if(a==5)
        {
            continue
        }
        document.write(a)
    }
}
</script>

```

In the above code, the series from 1 to 4 will be printed, the loop will break at 5, and again the series from 6 to 10 will be printed.

The with statement: The with statement helps the users to write shorter code related to an object. The properties and methods of that object can be invoked without writing the lengthy object reference. This can help to reduce the file size and also to check and debug the code.

The syntax of using the with statement is as follows:

```

with(object)
{
    //statements
}

```

The following example will demonstrate the use of the with statement:

```

<script language="javascript">
function mywith()
{
    with(document.form1)
    {
        elements[0].value="text1"
        elements[1].value="text2"
        elements[2].value="text3"
        elements[3].value="text4"
    }
}

```

</script>

The switch statement: The switch statement can be used for writing code that make decisions. Decision-making process is used to execute a program according to one or more conditions.

The syntax of using the switch statement is as follows:

```
switch(expression)
{
    case label:
        statements
        [break]
    case label:
        statements
        [break]
    [default
        statements
        break]
}
```

In the above syntax, the break statement and the default statement are optional. The break statement stops the execution of the switch statement after the execution of a case statement. If the break statement is removed, all the case statements following the correct case statement will be executed.

The following code will demonstrate the use of the switch statement:

```
<script language="javascript">
function myswitch()
{
    var a=prompt("Enter the letters a,b,c")
    switch(a)
    {
        case "a":
            alert("a is typed")
            break
        case "b":
            alert("b is typed ")
            break
        case "c":
            alert("c is typed ")
            break
        default:
            alert("It is a wrong character.")
            break
    }
}
</script>
```