Conditional Statement in Javascript

JavaScript supports the following forms of **if..else** statement −

- if statement
- if...else statement
- if...else if... statement.

→

```
if (expression) {
   Statement(s) to be executed if expression is true
               }
```

→

```
if (expression) {
   Statement(s) to be executed if expression is true
} else {
   Statement(s) to be executed if expression is false
               }
```

→

```
if (expression 1) {
   Statement(s) to be executed if expression 1 is true
} else if (expression 2) {
   Statement(s) to be executed if expression 2 is true
} else if (expression 3) {
   Statement(s) to be executed if expression 3 is true
} else {
   Statement(s) to be executed if no expression is true
                 }
```

Example

```
<html>
<head>
	<script type="text/javascript">
		var one = prompt("Enter the first number");
		var two = prompt("Enter the second number");
		one = parseInt(one);
		two = parseInt(two);
		if (one == two)
			document.write(one + " is equal to " + two + ".");
		else if (one<two)
			document.write(one + " is less than " + two + ".");
		else
			document.write(one + " is greater than " + two + ".");
	</script>
</head>
<body>
</body>
</html>
```

**** The **prompt**() method displays a dialog box that **prompts** the visitor for input.
A **prompt** box is often used if you want the user to input a value before entering a page. Note: When a **prompt** box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

**** The **parseInt()** function is used to accept the string ,radix parameter and convert it into an integer.

```
<html>  <head>  <body>
<script>
   var v1 = parseInt("3.14");

   document.write('Using parseInt("3.14") = '
   + v1 + "<br>");
</script> </body></head></html>
```

The **switch case** statement in JavaScript is also used for decision making purposes. In some cases, using the switch case statement is seen to be more convenient over if-else statements. Consider a situation when we want to test a variable for hundred different values and based on the test we want to execute some task. Using if-else statement for this purpose will be less efficient over switch case statements and also it will make the code look messy.
The switch case statement is a multiway branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression.

```
switch (expression)
{
   case value1:
      statement1;
      break;
   case value2:
      statement2;
      break;
   .
   .
   case valueN:
      statementN;
      break;
   default:
      statementDefault;
}
```

- *expression* can be of type numbers or strings.
- Dulplicate *case* values are not allowed.
- The *default* statement is optional. If the expression passed to switch does not matches with value in any case then the statement under default will be executed.
- The *break* statement is used inside the switch to terminate a statement sequence.
- The *break* statement is optional. If omitted, execution will continue on into the next case.

EXAMPLE :

```
var text;
var fruits = “Apple”;

switch(fruits) {
  case "Banana":
    text = "Banana is good!";
    break;
  case "Orange":
    text = "I am not a fan of orange.";
    break;
  case "Apple":
    text = "How you like them apples?";
    break;
  default:
    text = "I have never heard of that fruit...";
}
```

EXAMPLE:

```
<html>
  <body>
    <script type = "text/javascript">
      <!--
         var grade = 'A';
         document.write("Entering switch block<br />");
         switch (grade) {
           case 'A': document.write("Good job<br />");
           break;

           case 'B': document.write("Pretty good<br />");
           break;

           case 'C': document.write("Passed<br />");
           break;

           case 'D': document.write("Not so good<br />");
           break;

           case 'F': document.write("Failed<br />");
           break;
```

```
         default:  document.write("Unknown grade<br />")
      }
      document.write("Exiting switch block");
   //-->
  </script>
  <p>Set the variable to different value and then try...</p>
 </body> </html>
```

The JavaScript while statement creates a loop that executes a block of code as long as the test condition evaluates to true.
The following illustrates the syntax of the while statement.

```
while (expression) {
   Statement(s) to be executed if expression is true
                    }
```

The while statement evaluates the expression before each iteration of the loop.
If the expression evaluates to true, the while statement executes the statement. If the expression evaluates to false, execution continues with the statement after the while loop.
The while loop evaluates the expression before each iteration, therefore, the while loop is known as a pretest loop. For this reason, it is possible that the statement inside the while loop is never executed.

```
<html>
   <body>

      <script type = "text/javascript">
         <!--
            var count = 0;
            document.write("Starting Loop ");

            while (count < 10) {
               document.write("Current Count : " + count + "<br
/>");
               count++;
            }

            document.write("Loop stopped!");
         //-->
      </script>

      <p>Set the variable to different value and then try...</p>
   </body>
</html>
```

The '**for**' loop is the most compact form of looping. It includes the following three important parts −

- The **loop initialization** where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.
- The **test statement** which will test if a given condition is true or not. If the condition is true, then the code given inside the loop will be executed, otherwise the control will come out of the loop.
- The **iteration statement** where you can increase or decrease your counter.

```
for (initialization; test condition; iteration statement) {
   Statement(s) to be executed if test condition is true
}
```

```
<html>
   <body>
      <script type = "text/javascript">
         <!--
            var count;
            document.write("Starting Loop" + "<br />");

            for(count = 0; count < 10; count++) {
               document.write("Current Count : " + count );
               document.write("<br />");
            }
            document.write("Loop stopped!");
         //-->
      </script>
      <p>Set the variable to different value and then try...</p>
   </body>
</html>
```

For...in to be discussed later.