[redacted]

Web Development Programming Language

By: Dr. Galal AL-Marzoqi

[redacted]

Application of java script

JavaScript is most commonly used as a client-side scripting language. This means that JavaScript code is written into an HTML page. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it's up to the browser to do something with it.

Syntax: `<script>  code   </script>`

JavaScript Advantages and Disadvantages

Advantages	Disadvantages
- The JavaScript having an ability to produce the same result on all modern browsers. ***Ex:*** *When we write code in JavaScript and Use chrome, Firefox, opera, etc. the browsers show same result.* - Client-Side execution.	- Client-side Security. - Lack of Correcting Ability.

- **JavaScript Display Output**

JavaScript can "display" data in different ways:

- Writing into the HTML output using **document.write().**

✓ **Example**

```
<html>
 <head>
  <h>My First Web Page</h>
 </head>
 <body>
  <script>
      document.write("I am JavaScript")
  </script>
 </body>
</html>
```

✓ **Example**

```
<html>
 <head>
  <h>My First Web Page</h>
 <script>
  document.write("I am JavaScript")
  </script>
 </head>
</html>
```

Note: The JavaScript, we can write code in any place inside Notepad, but the browser read code line by line.

• Data Types and Variables

✓ **JavaScript data types**: numbers, strings and objects which present below:

```
var length = 16;                                   // Presents Number
var lastName = "Johnson";                          // Presents String
var x = {firstName:"John", lastName:"Doe"};        // Presents Object
```

Example in Number:

```
<html>
   <body>
      <h2>JavaScript</h2>
     <script>
         var x = 5;
          document.write (x)
     </script>
   </body>
</html>
```

✓ **Example in String:**

```
<html>
 <body>
     <h2>JavaScript</h2>
      <script>
       var x = "Volvo";
       document.write(x)
      </script>
 </body>
</html>
```

✓ **Example in Object:**

```
<html>
 <body>
      <h2>JavaScript</h2>
        <script>
var person = { firstName : "sam", lastName  : "Ali",};
document.write (person.firstName + " " + person.lastname + " ");
    </script>
 </body>  </html>
```

✓ **JavaScript variables** are containers for storing data values. In this example, x, y, and z, are variables:

✓ **Example in Addition:**

```
<html>
 <head>
  <h2>JavaScript Variables</h2>
 </head>
 <body>
  <script>
     var x = 5;    var y = 6;      var z = x + y;
     document.write(" This result:" + z)
  </script>
 </body> </html>
```

✓ **Example in Minus:**

```
<html>
 <head>
  <h2>JavaScript Variables</h2>
 </head>
 <body>
      <script>
       var x = 5;     var y = 6;     var z = y - x;
      document.write(" This result:" + z)
       </script>
</body> </html>
```

- ## JavaScript If statements

The **if/else** statement executes a block of code if a specified condition is true. If the condition is false, another block of code can be executed. The if/else statement is a part of JavaScript's "Conditional" Statements, which are used to perform different actions based on different conditions.

➢ In JavaScript has the following conditional statements:

- Use if to specify a block of code to be executed, if a specified condition is true
- Use else to specify a block of code to be executed, if the same condition is false
- Use else if to specify a new condition to test, if the first condition is false
- Use switch to select one of many blocks of code to be executed

JavaScript statements are collected of Values, Operators, Expressions, Keywords, and Comments. These statements tell the browser to write any things such as "Hello Dolly” or any things inside an HTML element.

➢ Syntax

1. The **if** statement specifies a block of code to be executed if a condition is true:

```
if (condition)
{
  // block of code to be executed if the condition is true
}
```

2. The **else** statement specifies a block of code to be executed if the condition is false:

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

3. The **else if** statement specifies a new condition if the first condition is false:

```
 if (condition1) {
    // block of code to be executed if condition1 is true
   }
 else if (condition2) {
   // block of code to be executed if the condition1 is false and condition2 is true
   }
 else {
    // block of code to be executed if the condition1 is false and condition2 is
false
}
```

✓ **Example in If:**

```
<html>
 <body>
  <script>
       var time = 2;
       var greeting;
        if (time < 2) {
          greeting = "Good day";
                 }
            else {
          greeting = "Good evening";
         }
         document.write (greeting);
  </script>
 </body>
</html>
```

✓ **Example in If..else:**

```
<html>
 <body>
  <script>
         var time;
         var greeting;
          if (time >=90) {
               greeting = "A";
                }
                else if (time >=80) {
                greeting = "You Are:  B";}
                else if (time >=70) {
                greeting = "You Are:  C";
                else if (time >=60) {
                greeting = "You Are:  D"; }
                else if (time >=50) {
                greeting = "You Are:  E"; }
                else if (time <50) {
                greeting = "You Are:  Failure";
         }
         document.write (greeting);
       </script>
       </body>
       </html>
```

- ## JavaScript switch statements

Use the switch statement to select one of many code blocks to be executed.

Syntax

```
switch(expression) {
      case x:
            // code block
       break;
      case y:
           // code block
      break;
      default:
          // code block
       }
```

This is how switch statement works:

1. The switch expression is evaluated once.

2.The value of the expression is compared with the values of each case.

3.If there is a match, the associated block of code is executed.

Example

The **getDay()** method returns the weekday a number between 0 and 6. (Sunday=0, Monday=1, Tuesday=2 .). This example uses the weekday number to calculate the weekday name:

```
<html>
<body>
<script>
var day;
switch (new Date().getDay()) {
  case 0:
    day = "Sunday";
    break;
  case 1:
    day = "Monday";
    break;
  case 2:
    day = "Tuesday";  break;
  case 3:
     day = "Wednesday";
    break;
case 4:
    day = "Thursday";
    break;
case 5:
    day = "Friday";
    break;
  case  6:
    day = "Saturday";
}
document.write("Today is " + day);
</script>
</body>
</html>
```

• String Methods in JavaScript

JavaScript are methods and properties which available to primitive values, because JavaScript treats primitive values as objects when executing methods and properties.

1. String Length (length)
2.Finding a String in a string (IndexOf() and lastIndexOf())

• String Length (length)

The length property returns the length of a string:

```
<html>
       <body>
           <h2>JavaScript String Properties</h2>
         <script>
           var txt = "ABCDEFG";
           document.write("The Text Length:  ", txt.length);
         </script>
</body>
       </html>
```

• Finding a String in a string (IndexOf() and lastIndexOf())

The indexOf() method returns the index of (the position (from zero 0 is the first position in a string, 1 is the second, 2 is the third ...) of the first occurrence of a specified text in a string and The lastIndexOf() method returns the index of the **last** occurrence of a specified text in a string: Both methods accept a second parameter as the starting position for the search. The lastIndexOf() methods searches backwards (from the end to the beginning), meaning.

✓ Example in indexOf()

```
<html>
  <body>
   <h2>JavaScript String Methods</h2>
   <script>
   var text= "Please you locate";
   var result= text.indexOf("locate");
   document.write("Index locate is: ", result);
  </script>
 </body>
</html>
```

✓ Example in lastIndexOf()

```
<html>
  <body>
<h2>JavaScript String Methods</h2>
   <script>
     var text= "Please you locate";
     var result = text.lastIndexOf("you");
     document.write("Index locate is: ", result);
   </script>
  </body>
</html>
```