

Experiment No. 04

Title: Write a JavaScript to compute mathematical operations on Client side.

Objective: To study JavaScript functions.

Theory:

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

Client-Side JavaScript -

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI serverside scripts. For example, you might use JavaScript to check if the user has entered a valid email address in a form field. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server. JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

JavaScript Syntax -

JavaScript can be implemented using JavaScript statements that are placed within the `<script>... </script>` HTML tags in a web page. You can place the `<script>` tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the `<head>` tags. The `<script>` tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows-

```
<script>  
    JavaScript Code  
</script>
```

The script tag takes two important attributes –

Language – This attribute specifies what scripting language you are using. Typically, its value will be JavaScript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.

Type – This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

So JavaScript segment look like-

```
<script language="javascript" type="text/javascript">  
    JavaScript Code  
</script>
```

First JavaScript Code:

```
<html>  
  <head>  
    <script language = "javascript" type = "text/javascript">  
      <!--  
        document.write("Hello World!")  
      //-->  
    </script>  
  </head> </html>
```

In above code function document.write() which writes a string into our HTML document.

JavaScript Functions-

The most common way to define a function in JavaScript is by using the function keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

Syntax:

```
<script language = "javascript" type = "text/javascript">  
  function functionname(parameter-list)  
  {  
    statements  
  }  
</script>
```

JavaScript Event-

JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page. When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

onclick Event Type

This is the most frequently used event type which occurs when a user clicks the left button of his mouse. You can put your validation, warning etc., against this event type.

Example:

```
<html> <head>
  <script type = "text/javascript">
    <!--
      function sayHello() {          alert("Hello World");
    }
    //-->
  </script>
</head>
<body>
  <p>Click the following button and see result</p>
  <form>
    <input type = "button" onclick = "sayHello()" value = "Say Hello" />
  </form>
</body>
</html>
```

onsubmit Event Type:

The onsubmit is an event that occurs when you try to submit a form. You can put your form validation against this event type.

Example:

The following example shows how to use onsubmit. Here we are calling a validate() function before submitting a form data to the webserver. If validate() function returns true, the form will be submitted, otherwise it will not submit the data.

```
<html>
<head>
  <script type = "text/javascript">
    <!--
      function validation() {

        all validation goes here
        .....
        return either true or false
      }
    //-->
  </script>
</head>
```

```
<body>
  <form method = "POST" action = "t.cgi" onsubmit = "return validate()">
    .....
    <input type = "submit" value = "Submit" /> </form> </body></html>
```

Algorithm:

1. Design html form with multiple textbox and button.
2. Add event handler with onclick event
3. Call add function on onclick event
4. Perform various arithmetic operations
5. Display Result

Key Concept: Html form, JavaScript, Event