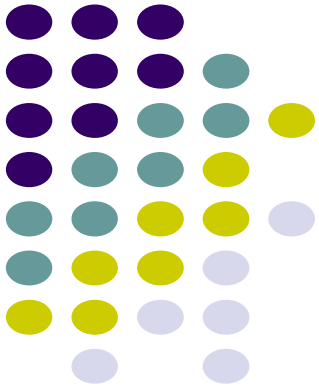
Introduction to JavaScript

Dr. Arul Xavier V M
Assistant Professor



Introduction

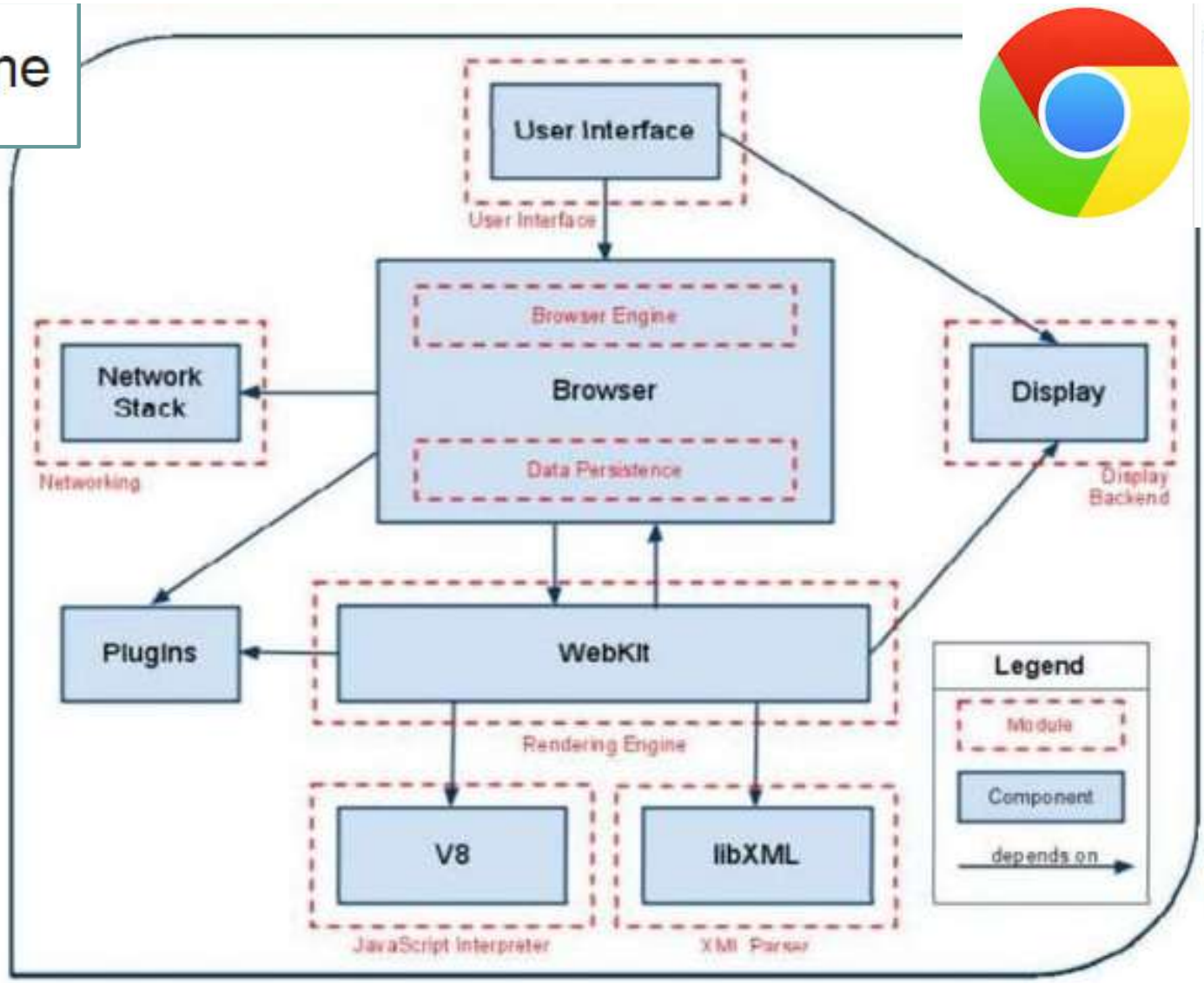
- JavaScript is the world's most popular programming language.
- JavaScript is *an* ***object-based scripting language*** that is lightweight and cross-platform.
- **JavaScript** for beginners and professionals to create interactive client side dynamic pages.
- JavaScript is not compiled but translated.
- The JavaScript Engine (**embedded in browser**) is responsible to translate the JavaScript code.

JavaScript engine

- A JavaScript engine is a **program or interpreter** which executes JavaScript code.
- A JavaScript engine may be a **traditional interpreter.**
- Every browser has an in built Javascript engine. Popular Javascript names are given below.
  - Google – V8
  - Firefox – SpiderMonkey
  - Safari – JavaScriptCore

Google Chrome

Web browser



History of JavaScript

- JavaScript was created in 10 days in May 1995 by **Brendan Eich.**
- The original name was ***"Mocha",*** and then changed to ***"LiveScript".***
- Later, upon receiving a trademark license from Sun Microsystems, the name **JavaScript** was adopted.

JavaScript vs Java

- Javascript is not a Java
- It is not a light version of Java
- It was not based on Java
- It does not matter if you know Java
- Note:
  - Javascript is not all related to Java Programming Language

How to include JavaScript code in HTML?

- HTML provides 2 places to put your JavaScript code:
  - Internal JavaScript
    - Using <script> tag
      - Inside <head> tag or
      - Inside <body> tag
  - External JavaScript file.
    - Using external file with extension ".js".
    - Then include the file using <script> tag

Internal JavaScript

- You can include the JavaScript code internally by embedding using <script> tag in your HTML page.
- You can place the <script> tag either inside the <head> tag, or inside the <body>.

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Website</title>
        <script>
            <!-- Javascript code goes here -->
        </script>
    </head>
    <body>
        <script>
            <!-- Javascript code goes here -->
        </script>
    </body>
</html>
```

External Javascript

- Scripts can also be placed in external files.
- External scripts are practical when the same code is used in many different web pages.
- JavaScript files have the file extension .js.
- To use an external script, put the name of the script file in the src (source) attribute of a <script> tag.

External Javascript

- Create a .js file and keep some Javascript code

myscript.js

```
1  /* Java Script Code Goes Here */
```

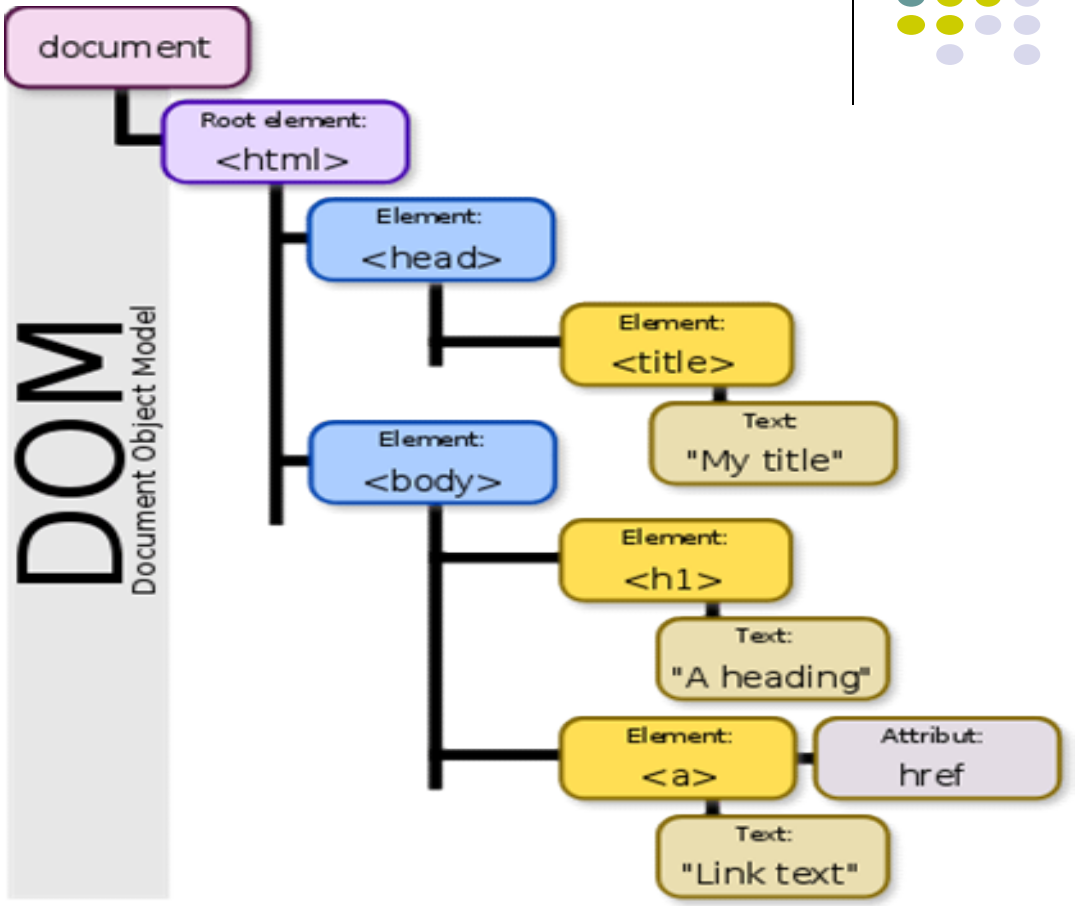
Link external Javascript file in HTML

- Link the external .js File in HTML code via
  - <script src= “myscript.js” >

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Website</title>
           <script src="myscript.js"> </script>
    </head>
    <body>

    </body>
</html>
```

Document Object Model (DOM)

- When a web page is loaded, the browser creates a Document Object Model of the page.
- The HTML DOM model is constructed as a tree of Objects:



Use of DOM with Javascript

- With the object model, JavaScript gets all the power it needs to create dynamic HTML:
- JavaScript can access, change, add or remove
  - HTML Elements or Tags
  - HTML Attributes
  - HTML Events
  - React to HTML Events
  - CSS Styles

Working with JavaScript

- JavaScript is object based language.
- In JavaScript, functions, events and properties are major elements to make the web page more interactive.
- There 2 variants of functions
  - Built-in functions
  - User-defined function
- JavaScript functions are also called as “Methods” (Both are same)

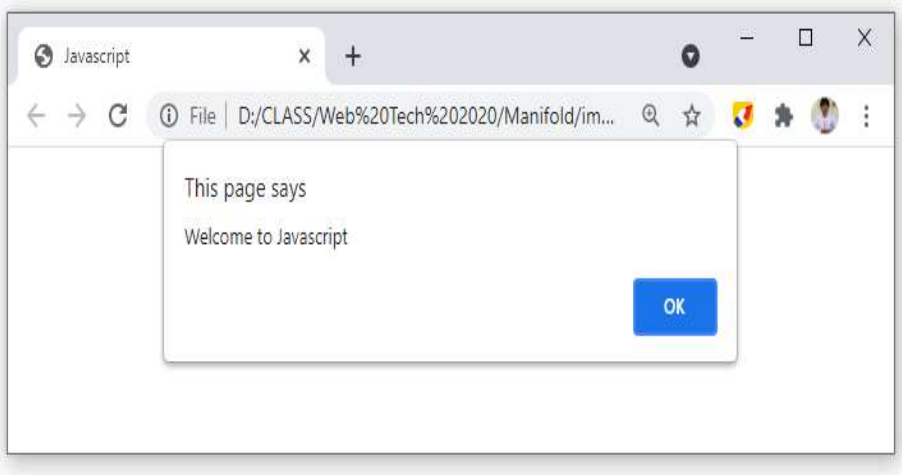
Display Data in Javascript

- JavaScript can "display" data in different ways using its built-in functions/methods.
  - Display using an alert box, using window.alert().
  - Display inside web page using document.write().
  - Display in browser console, using console.log().
  - Display inside HTML element using innerHTML property.
  - Display inside HTML Form Tex Box using value property.

To display data in a Alert or Message box

- The alert() method displays an alert box with a specified message and an OK button.
- The alert() method is defined by window object

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Website</title>
        <script>
            alert("Welcome to Javascript");
        </script>
    </head>
    <body>

    </body>
</html>
```

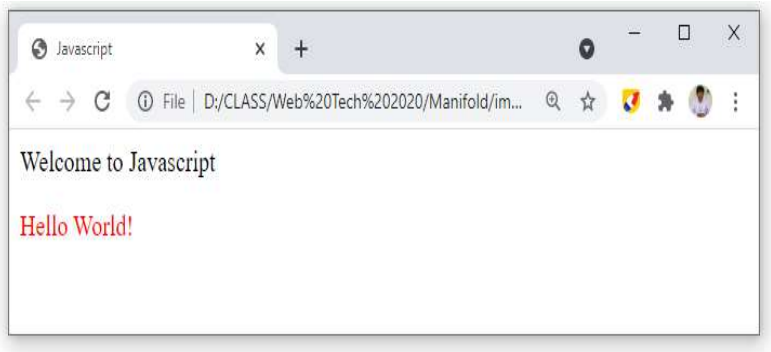


Display data into the Web Page.

- The HTML DOM model provides an object called "document", which includes set of methods.
- One of the method is write(), which can be used to display any data inside the web page using Javascript.

Syntax: document.write()

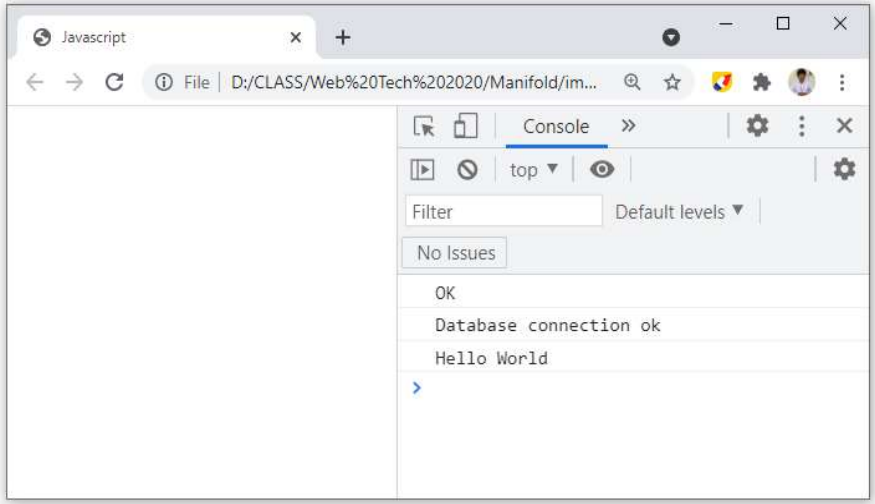
```
<!DOCTYPE html>
<html>
    <head>
        <title>My Website</title>
        <script>
      document.write("Welcome to Javascript");
      document.write("<p style='color:red;'>Hello World!</p>");
        </script>
    </head>
    <body>

    </body>
</html>
```



Display data in a Browser's Console Window

- For debugging purposes, you can call the console.log() method in the browser to display data.

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Website</title>
        <script>
       console.log("OK");
       console.log("Database connection ok");
       console.log("Hello World");
        </script>
    </head>
</html>
```



Display inside HTML element, using innerHTML property.

- JavaScript is a object-based language.
- Every HTML element is consider as an “object”.
- “**object**” consists of “**properties**” and “**methods**”.
  - innerHTML is a one of the property of HTML elements(limited!!)
  - It is used to write data to specific element’s content area directly.
- To access any HTML element, JavaScript can use the following method

  document.getElementById(**id**);
- The **id** attribute defines a unique identity of the HTML element.
- The **innerHTML** property defines the content of the particular element.

HTML5Application - Apache NetBeans IDE 12.2

File Edit View Navigate Source Refactor Run Debug Team Tools Window Help

index.html

```
6        <meta name=                                  initial-scale=1.0">
7
8    </head>
9    <body>
10       <div id="bo
11           Static
12       </div>
13        <script>
             document.getEl
15       </script>
16   </body>
17 </html>
18
```

document.getElementById(elementId)

Returns the Element that has an ID attribute with the given value. If no such element exists, this returns null. If more than one element has an ID attribute with that value, what is returned is undefined. The DOM implementation is expected to use the attribute Attr.isId to determine if an attribute is of type ID. Note: Attributes with

- getElementById(elementId: String): Eleme... JS Platform
- getElementsByClassName(names: String): N... JS Platform
- getElementsByName(elementName: String): ... JS Platform
- getElementsByTagName(tagname: String): N... JS Platform
- getElementsByTagNameNS(namespaceURI: Str... JS Platform

html > body > script

14:27 INS Unix (LF)

10:16 AM 2/16/2021

Example

```
<!DOCTYPE html>
<html>
   <head>
      <title>Javascript</title>
   </head>
   <body>
      <div id="box1" style="color:blue;"></div>
      <script>
         document.getElementById('box1').innerHTML = "Welcome User!!";
       </script>
   </body>
</html>
```

Closer look at document.getElementById()…

Method

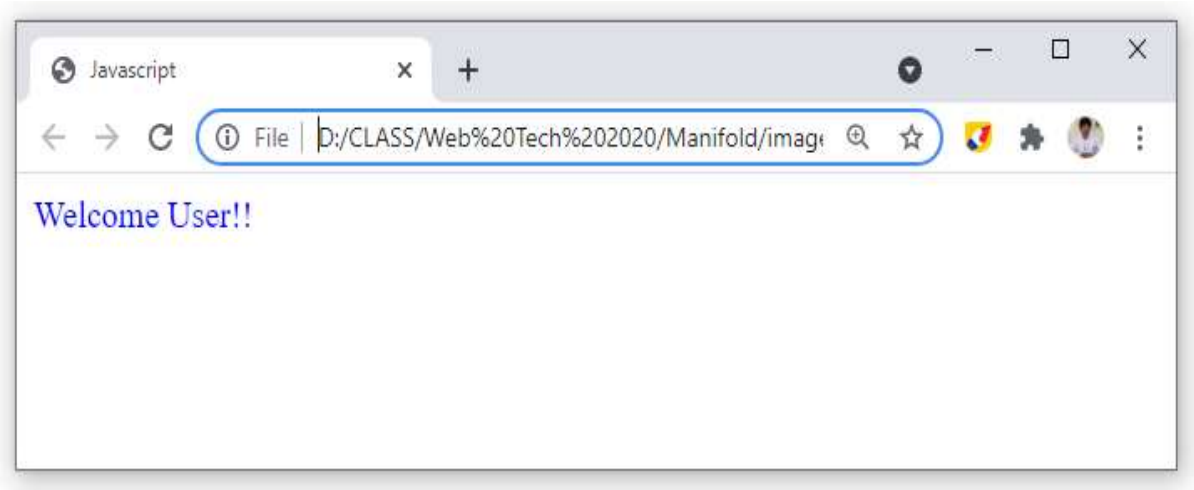
Property to access the content area of DIV element

```
document.getElementById('box1').innerHTML = "Welcome User!!";
```

DOM object

id of div Element

data



Display inside a HTML Form Text Box

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    Text Box: <input type="text" id="output">
    <script>
        document.getElementById('output').value = "100";
    </script>
</body>
</html>
```

Javascript Programming Features

- JavaScript is a programming language:
- All instructions are called as statements, which must be separated by semicolon.
- JavaScript statements are composed of:
  - Values(literals), Variables, Operators, Expressions, Control Statements, Keywords, and User Defined Functions
  - Values or literals are: Numbers, Strings, Arrays etc..
  - Variables can be used represent Numbers, Strings, Arrays, Objects, etc.

Creating variables using var keyword

```
<script>
  var price=1000;
  var name="Prince";
  var a,b,c;
  var interest=7.5;
</script>
```

String Concatenation

- The + operator can also be used to add (concatenate) strings.

```
<body>
    <div id="box" style="color: blue;font-size: 30px;"></div>
    <script>
        var firstname = 'Arul '
        var lastname ='Xavier'
        var name = firstname + lastname;
        document.getElementById('box').innerHTML = name;
    </script>
</body>
```

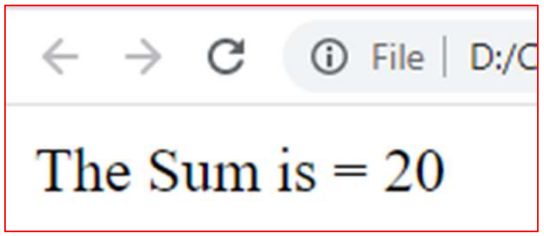
Creating user defined Function

- Functions are created using the keyword function.
- The code to be executed, by the function, is placed inside curly brackets: {}
- Function parameters are listed inside the parentheses () in the function definition.

```
function name(parameter1, parameter2, parameter3)
{
  // code to be executed
}
```

Creating Function - Example

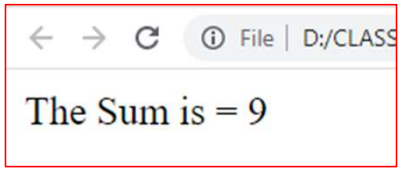
```
<script>
    function add(){        // creating a function
        var a = 10;
        var b = 10;
        sum = a+b;
        document.write("The Sum is = " + sum);
    }

    add();       //calling or invoking the function

</script>
```



Adding Parameters to function

- The parentheses may include parameter names separated by commas: (parameter1, parameter2, ...)
- Function arguments are the values received by the function when it is invoked.
- Inside the function, the arguments (the parameters) behave as local variables.

```
<script>
    function add(num1, num2){    // creating function with parameters
        var sum = num1+num2;
        document.write("The Sum is = " + sum);
    }

    add(5,4); //calling or invoking the function with arguments

</script>
```

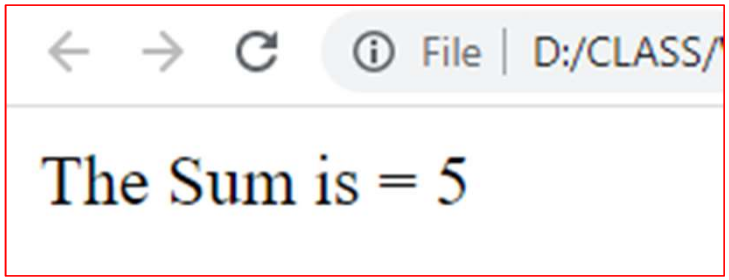


Return a value from a Function

- When JavaScript reaches a return statement, the function will stop executing.
- If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.
- Functions often compute a return value.
- The return value is "returned" back to the "caller" code

Return a value from a Function - Example

```
<script>
    function add(num1, num2)   // creating function with parameters
    {
        return num1+num2;
    }

    result=add(5,4);   // the value will be return to result variable

    document.write("The Sum is = " + result);
</script>
```



Calling Functions

```
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <script>
            show();
            function show(){
                document.write("Function calling..");
            }
        </script>
    </head>
    <body>
        <div>This is a HTML Content..</div>
    </body>
</html>
```

Function calling from <body>

```
<!DOCTYPE html>
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <script>
            function show(){
                document.write("Function calling..");
            }
        </script>
    </head>

    <body>
        <div>This is a HTML Content..</div>
        <script>
            show();
        </script>
    </body>
</html>
```

This is a HTML Content..
Function Calling...

Event Handling in JavaScript

- HTML events are **“state changes"** that happen to HTML elements by browser or user.
- When JavaScript is used in HTML pages, JavaScript can **"react"** on these events.
- Examples of HTML events:
  - An HTML web page has finished loading
  - An HTML input field was changed
  - An HTML button was clicked

Common HTML Events

Event name	Description
onload	The browser has finished loading the page
onclick	The user clicks an HTML element
onchange	An HTML element has been changed
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pressing a keyboard key
onkeyup	The user releases a keyboard key
onfocus	The element gets the focus on it
onblur	The event occurs when an element loses focus
oninvalid	The event occurs when an element is invalid
onselect	The event occurs after the user selects some text (for <input> and <textarea>)
onsubmit	The event occurs when a form is submitted

Writing Event in HTML

With single quotes:

```
<element event='some JavaScript'>
```

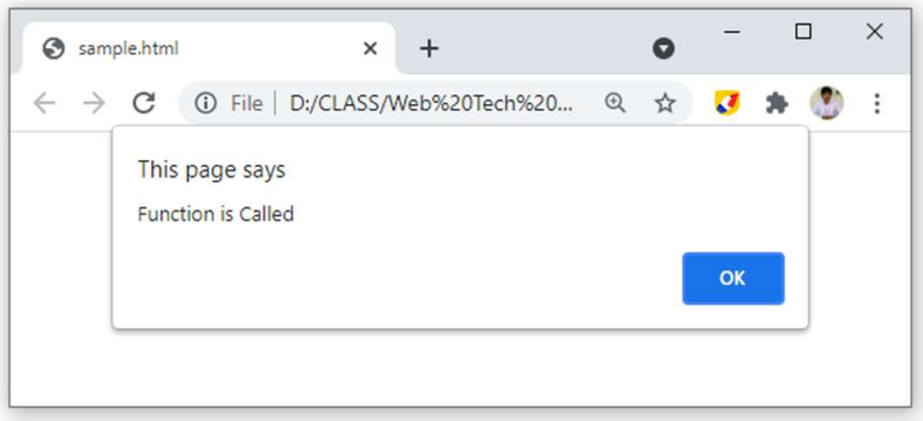
With double quotes:

```
<element event="some JavaScript">
```

Handling Event - onload

- onload event occurs whenever the browser has finished loading the page
- For example,
  - The alert function will be called when page loads.

```
<!DOCTYPE html>
<html>
   <head>
     <script>
     function test(){
         alert("Function is Called");
       }
    </script>
   </head>
   <body onload="test()">

   </body>
</html>
```



Calling Functions when a button is clicked

- onclick event occurs whenever a user clicks on a particular HTML element.

```
<html>
   <head>
  <script>
     function fun1(){
         alert("fun1 is called when 'Compute' button is clicked");
     }
     function fun2(){
         alert("fun1 is called when 'Convert' button is clicked");
     }
  </script>

   </head>
   <body>
      <input type="button" value="Compute" onclick="fun1()"> <br><br>
      <button onclick="fun2()">Convert</button>
   </body>
</html>
```

Display inside HTML Element on event

```html
<!DOCTYPE html>
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <script>
            function fun(){
                document.getElementById('box1').innerHTML = "Fun..";
            }
        </script>
    </head>
    <body>
        <div id="box1" style="color:blue;font-size: 20px;margin-left: 50px;">
        </div>
        <button onclick="fun()">Login</button>
    </body>
</html>
```

Reading values from Input Elements

Text Box Input Element

- To read the value from a text input fields, first we need to find the text box using its ID.
- Then, “value” property returns the value enter by the user.

HTML DOM object

Property to access the value text box entered by the user

```
var data=document.getElementById('name').value;
```

variable

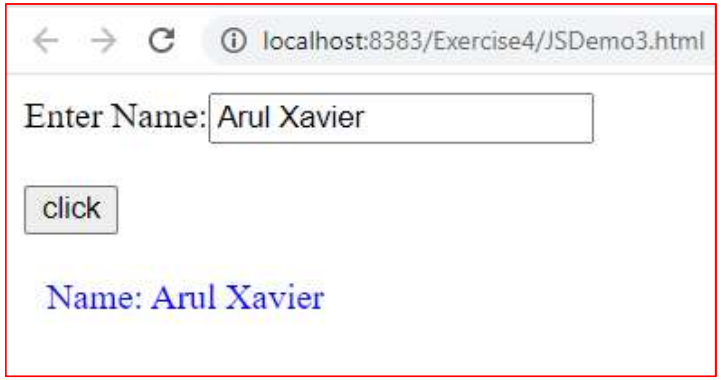
id of input Element

Example1: Read the data and display it

```html
<!DOCTYPE html>
<html>
    <head>
        <script>
            function read(){
                var name=document.getElementById('t1').value;
                document.write("Name: "+name);
            }
        </script>
    </head>
    <body>
        Enter Name:<input type="text" id="t1"><br><br>
        <button onclick="read()">click</button><br><br>
    </body>
</html>
```

Example 2: Read data and display inside HTML Element

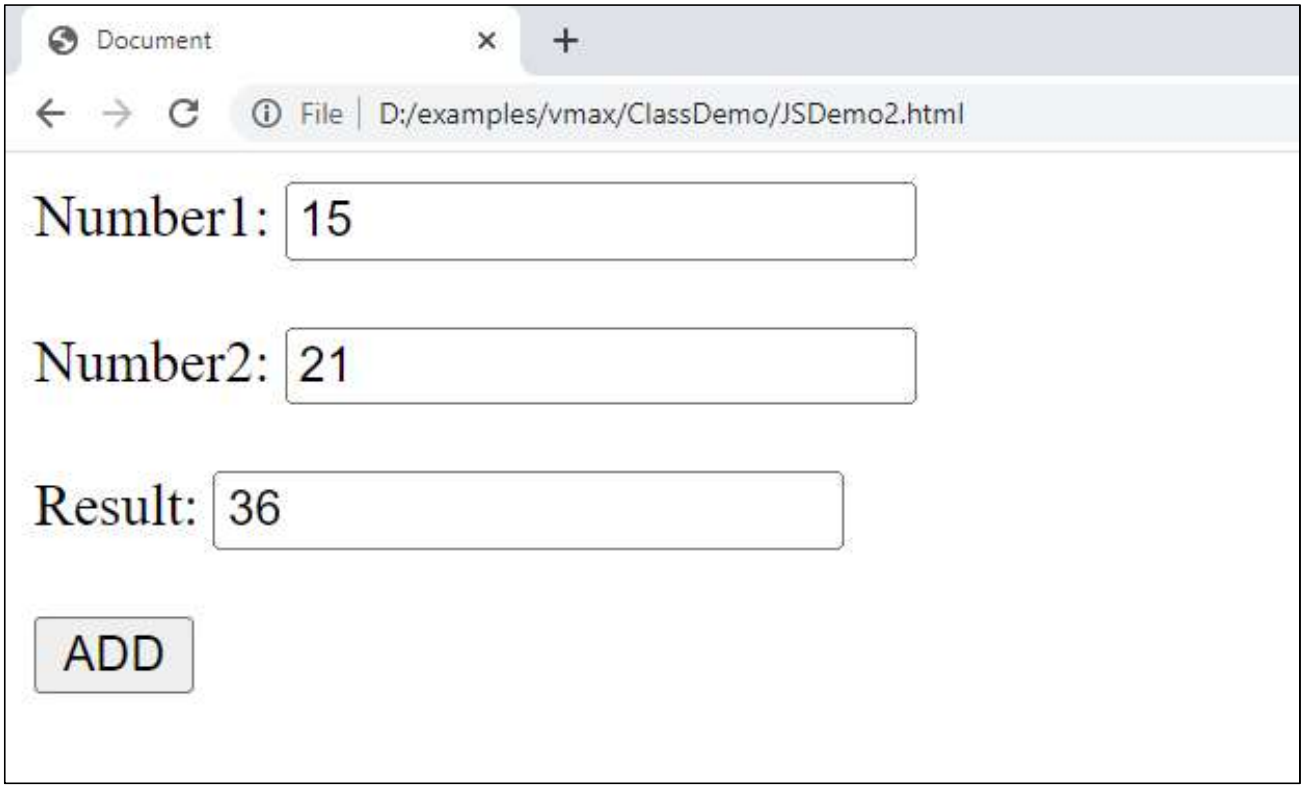
```html
<!DOCTYPE html>
<html>
    <head>
        <script>
            function read(){
                var name=document.getElementById('t1').value;
                document.getElementById('box1').innerHTML = "Name: "+name;
            }
        </script>
    </head>
    <body>
        Enter Name:<input type="text" id="t1"><br><br>
        <button onclick="read()">click</button><br><br>
        <div id="box1" style="color:blue;margin-left: 10px;">
        </div>
    </body>
</html>
```



Example: Add Two Numbers

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script>
        function add(){
            var a = parseInt(document.getElementById('tb1').value);
            var b = parseInt(document.getElementById('tb2').value);
            var sum = a+b;
            document.getElementById('result').value = sum;
        }
    </script>
</head>
<body>
      Number1: <input type="text" id="tb1"><br><br>
      Number2: <input type="text" id="tb2"><br><br>
      Result: <input type="text" id="result"><br><br>
      <button onclick="add()">ADD</button>
</body>
</html>
```

Example: Add Two Numbers



Exercise 1 - onclick

Develop a HTML program using JQuery to perform Simple Interest Calculation as given below.
[Formua: A = P *(1+r*t), A – Amount, P – Principle Amount, r- interest rate, t – tenure]

Simple Interest Calculator

Principle Amount	10000
Annual Interest Rate(%):	10
Tenure(years)	5

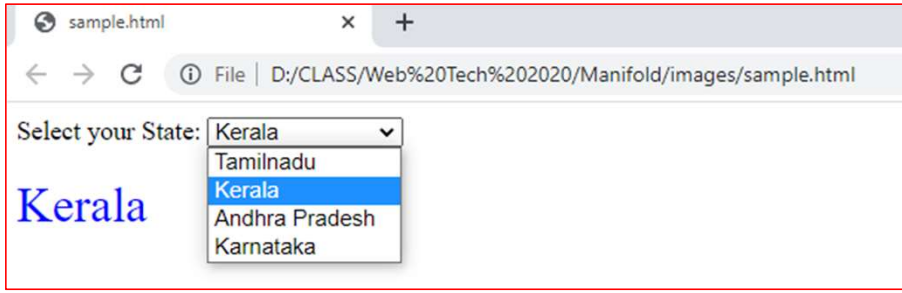
Calculate

Final Amount: Rs. 15000
Total Interest: Rs.5000

onchange event

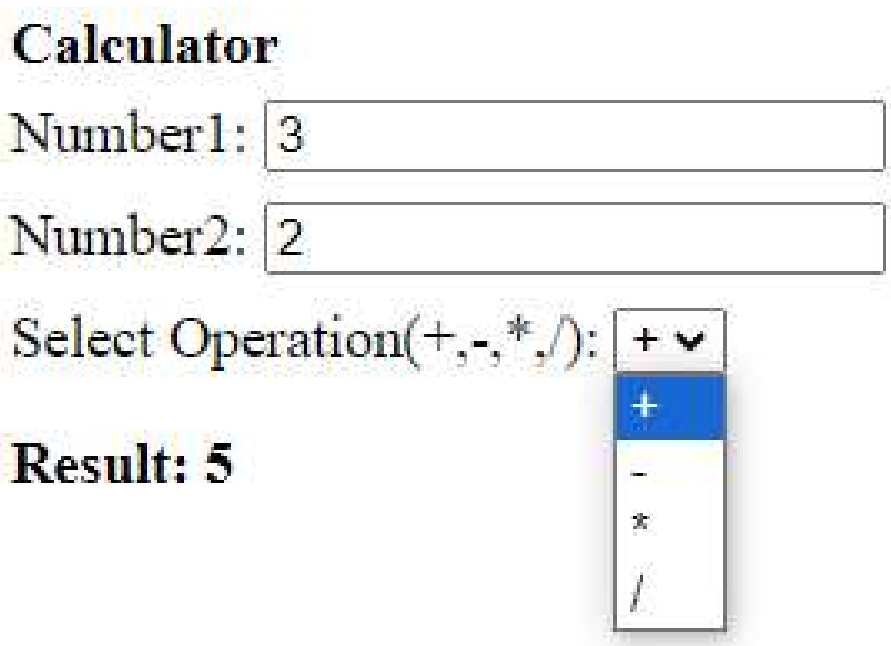
- The onchange event occurs when the value of an element has been changed.

```
<html>
   <head>
    <script>
      function test(){
         var data=document.getElementById('state').value;
         document.getElementById('output').innerHTML = data;
       }
   </script>
   </head>
   <body>
      Select your State:
      <select id="state" onchange="test()">
         <option>Tamilnadu</option>
         <option>Kerala</option>
         <option>Andhra Pradesh</option>
         <option>Karnataka</option>
      </select>
      <br><br>
      <div id="output" style="color:blue;font-size:30px;"></div>
   </body>
</html>
```



Exercise 1 – onchange event

Develop a simple mathematical calculator as given below using HTML5 and JavaScript.



Exercise 3

Design a HTML web page with JavaScript code to perform the Electricity Bill (EB) calculation as given below

- Get user details including name, previous unit and current unit in a HTML input fields.
- Calculate the consumed units and perform EB calculation with following constraints when the "Generate Bill" button is clicked.

  0-100 units = Free
  101-400 units = Rs. 4.50 per unit
  401- 500 units = Rs. 6 per unit
  501 and above units= Rs. 8 per unit
- Display the customer name, total units consumed and bill amount using DIV element.

Exercise 4

Create an HTML program with JavaScript to calculate and display the BMI value and Status for the given weight and heights.

BMI = weight(kg)/[height(m)]^2

Check the status using the following conditions.

- Underweight: BMI < 18
- Normal: BMI between 18 – 25
- Overweight: BMI between 25 – 30
- Obese: BMI > 30

BMI Calculator

Weight(Kg): 95

Height(m): 180

Find BMI

BMI Value: 29.3

BMI Status: Overweight

Exercise 5

- Create an online registration form using HTML as given below with the necessary attributes to make all fields mandatory.

Name: Enter Username

Password: **********

Email: sample@gmail.com

Phone: 10 digits

Date of Birth: mm/dd/yyyy

Gender: ◉ **Male** ○ **Female**

Skills: ☑ **HTML** ☐ **C** ☑ **Java** ☐ **Python**

Select State Tamilnadu

save clear