Aror University of Art, Architecture, Design & Heritage Sukkur

---

Lab 15: Introduction of JavaScript

Subject: Application of Information & Communication Technologies

Instructor: Abdul Ghafoor Date: Dec 11, 2023

Introduction:

JavaScript is a dynamic, versatile programming language primarily used for client-side scripting in web development.

Created by Brendan Eich in 1995, JavaScript has evolved from its origins at Netscape and is unrelated to the Java programming language.

Versions:

- ES5
- ES6

Development Environment:

- Node.js: Extends JavaScript to server-side development.

Frameworks and Libraries:

- React: Facebook's UI library.
- Angular: Google's framework for large-scale applications.

JavaScript Can Change HTML Content

One of many JavaScript HTML methods is getElementById().

The example below "finds" an HTML element (with id="demo"), and changes the element content (innerHTML) to "Hello JavaScript":

Example

```
<!DOCTYPE html>
<html>
<body>
```

```
<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello
JavaScript!"'>Click Me!</button>

</body>
</html>
```

JavaScript Functions and Events

- A JavaScript function is a block of JavaScript code, that can be executed when "called" for.
- For example, a function can be called when an event occurs, like when the user clicks a button.
- You can place any number of scripts in an HTML document.
- Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

Example

```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>
<p id="demo"> You can change the html content using the html </p>

<button type="button" onclick="myFunc()"> click to change the content </button>

<script>
        function myFunc(){
                document.getElementById("demo").innerHTML="Hello JavaScript!!!";
        }
</script>

</body>
</html>
```

JavaScript Display Possibilities

JavaScript can "display" data in different ways:

- Writing into an HTML element, using innerHTML.
- Writing into the HTML output using document.write().
- Writing into an alert box, using window.alert().

Using innerHTML

- To access an HTML element, JavaScript can use the document.getElementById(id) method.
- The id attribute defines the HTML element. The innerHTML property defines the HTML content:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My First Paragraph</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```

Using document.write()

- For testing purposes, it is convenient to use document.write():

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

Using window.alert()

- You can use an alert box to display data:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
```

```
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

You can skip the window keyword.

In JavaScript, the window object is the global scope object. This means that variables, properties, and methods by default belong to the window object. This also means that specifying the window keyword is optional:

JavaScript Variables

Variables are Containers for Storing Data

JavaScript Variables can be declared in 4 ways:

- Automatically
- Using var
- Using let
- Using const

Automatically

In this first example, x, y, and z are undeclared variables.

They are automatically declared when first used:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Variables</h1>

<p>In this example, x, y, and z are undeclared.</p>
<p>They are automatically declared when first used.</p>

<p id="demo"></p>

<script>
x = 5;
y = 6;
z = x + y;
document.getElementById("demo").innerHTML =
```

```
"The value of z is: " + z;
</script>

</body>
</html>
```

Using var

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Variables</h1>

<p>In this example, x, y, and z are variables.</p>

<p id="demo"></p>

<script>
var x = 5;
var y = 6;
var z = x + y;
document.getElementById("demo").innerHTML =
"The value of z is: " + z;
</script>

</body>
</html>
```

JavaScript Arithmetic Operators

Arithmetic operators perform arithmetic on numbers (literals or variables).

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (ES2016)
/	Division
%	Modulus (Remainder)
++	Increment
--	Decrement

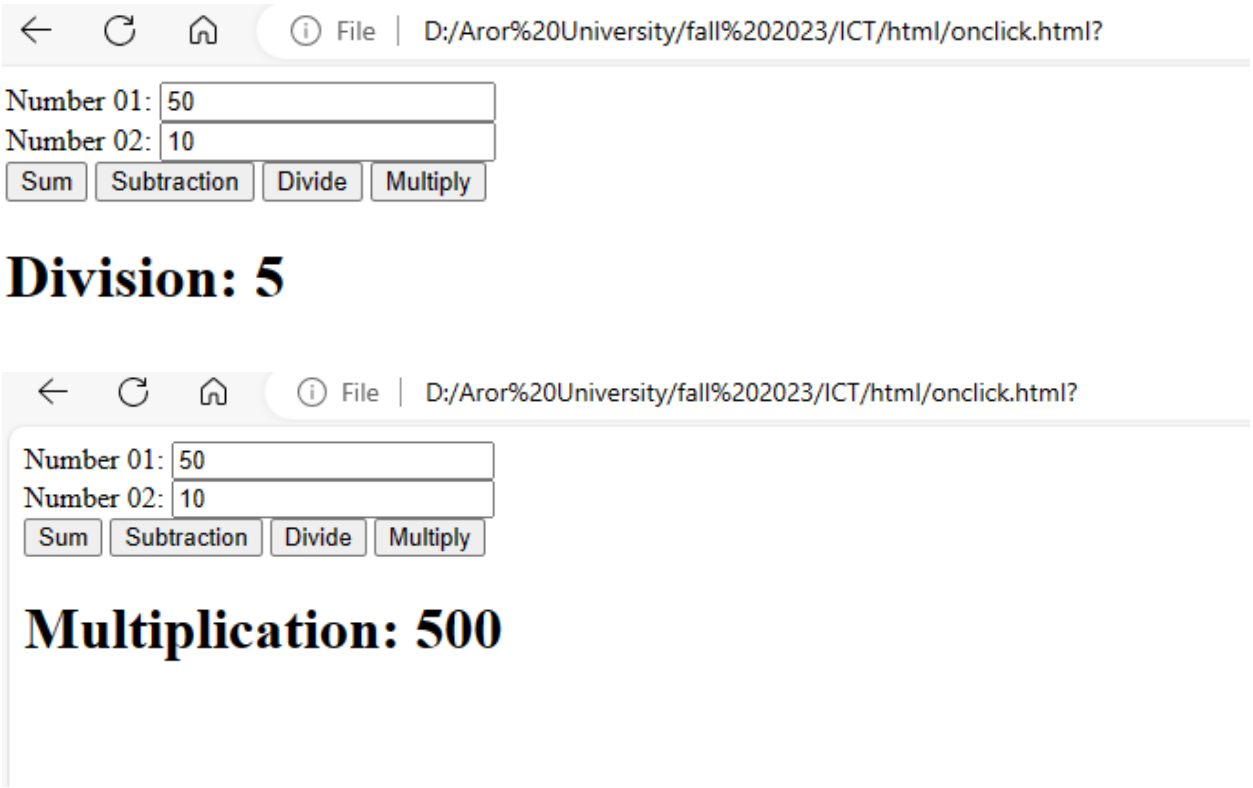
Exercise 01: Add Two Numbers

This program allows users to easily calculate the sum of two numbers. It consists of a simple interface with two input fields for entering numbers and a button to trigger the calculation using javaScript

```
<!DOCTYPE html>
<html>
<head>
 <script>
  function add() {
   event.preventDefault();
   var num1 = parseInt(document.getElementById("fn").value);
   var num2 = parseInt(document.getElementById("sn").value);
   if (!isNaN(num1) && !isNaN(num2)) {
    var sum = num1 + num2;
    document.getElementById("ans").textContent = "Sum: " + sum;
   } else {
    alert("Please enter valid numbers!");
   }
  }
 </script>
</head>
<body>
 <form>
  Number 01: <input type="text" id="fn"><br>
  Number 02: <input type="text" id="sn"><br>
  <button type="button" onclick="add()">Sum</button>
 </form>
 <h1 id="ans">.....</h1>
</body>
</html>
```

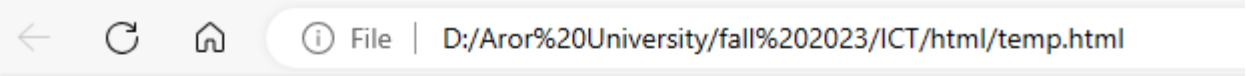
Output:

Task 01: Expand the exercise 01 by incorporating additional operations such as subtraction, multiplication, and division.



Task 02: Task: Celsius to Fahrenheit Converter

- Create a program that converts temperatures from Celsius to Fahrenheit.
- Implement the conversion using the formula: Fahrenheit (F) = Celsius (C) × 9/5 + 32.
- The program should take user input for the temperature in Celsius.
- Display the converted temperature in Fahrenheit.
- Ensure the program handles invalid input gracefully, providing a clear message to the user.
- Test the converter with various input values to verify its accuracy.



Celsius Temperature: 34
Convert to Fahrenheit

Temperature in Fahrenheit: 93.20 °F

Task 03: Decimal Number to Binary Number conversation

Hint:

In JavaScript, when using the toString() method on a number, the argument passed to toString() specifies the base of the numeral system to be used for the conversion. The most common bases are:

- toString(2): Binary (base 2)
- toString(8): Octal (base 8)
- toString(10): Decimal (base 10)
- toString(16): Hexadecimal (base 16)

(5).toString(2); output will: 101

Decimal to Binary Converter

Enter Decimal Number: 55 Convert to Binary

Binary representation: 110111