Tishk International University
IT Department
Course Code: IT 349/A

Web Programming

Week #2
Introduction to Javascript

Fall 2024
Hemin Ibrahim, PhD
hemin.ibrahim@tiu.edu.iq

Overview

- What is Javascript?
- JS roles in web development
- Application of JS
- Variables
- Operators
- Arrays
- Control statements
- Functions
- Objects
- DOM

Objectives

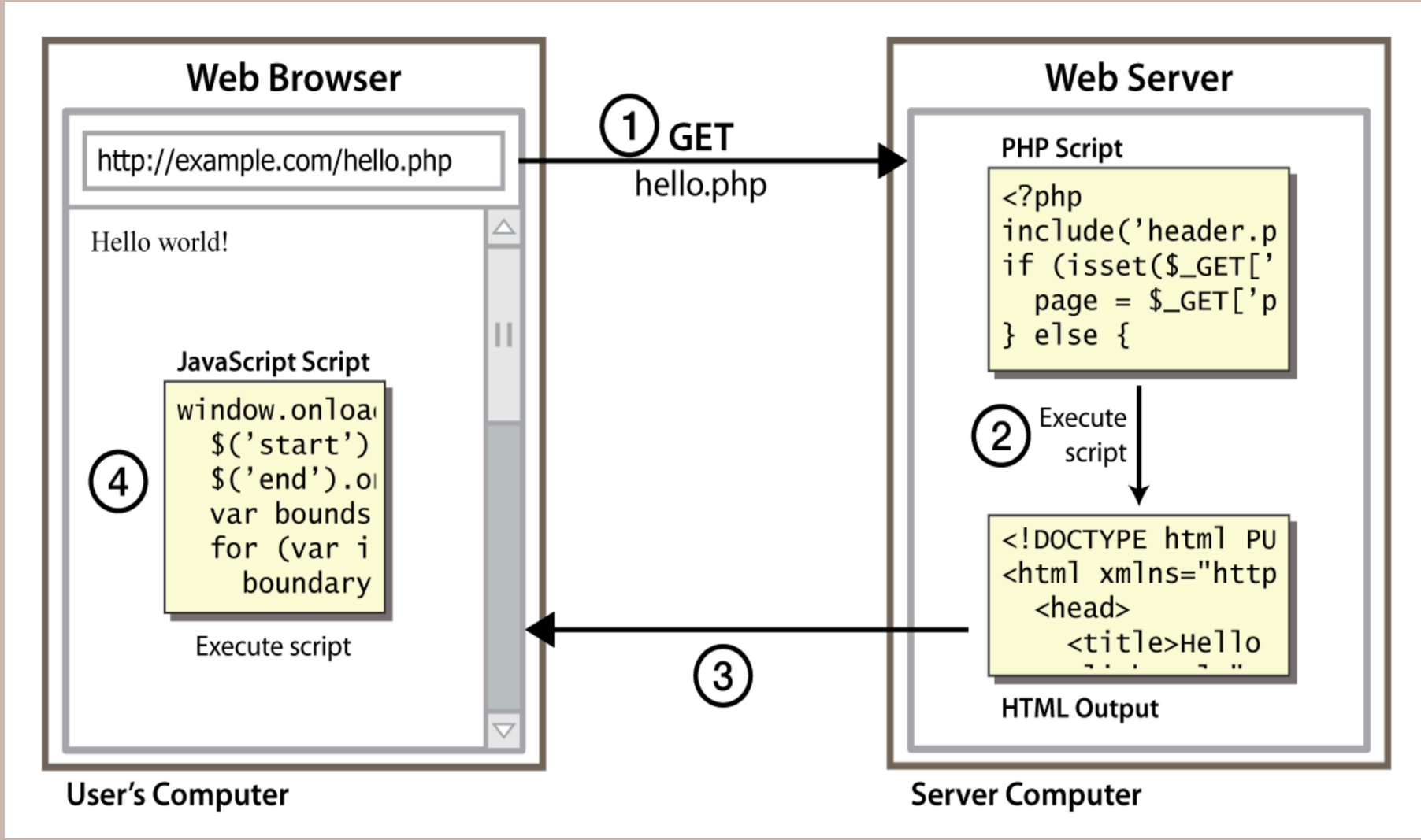
- Understand the Fundamentals of JavaScript
- Utilize Core JavaScript Syntax and Constructs
- Implement Control Flow in JavaScript Programs
- Develop Modular and Reusable Code with Functions and Objects
- Interact with Web Pages Using the Document Object Model (DOM)

What is Javascript?

- JavaScript is a high-level, interpreted programming language that enables interactive web pages. It is one of the core technologies of the World Wide Web, alongside HTML and CSS.

- HTML defines the structure of web content.
- CSS styles the appearance of content.
- JavaScript adds interactivity to web pages.

Role in Web Development



Role in Web Development

- **Client-Side Scripting**: Runs in the user's browser without needing server interaction.
- **Dynamic Content:** Update and change both HTML and CSS.
- **Event Handling:** Respond to user actions like clicks and key presses.
- **APIs and Libraries:** Access to numerous APIs (e.g., DOM, Fetch) and libraries/frameworks (e.g., React, Angular).

Why Javascript

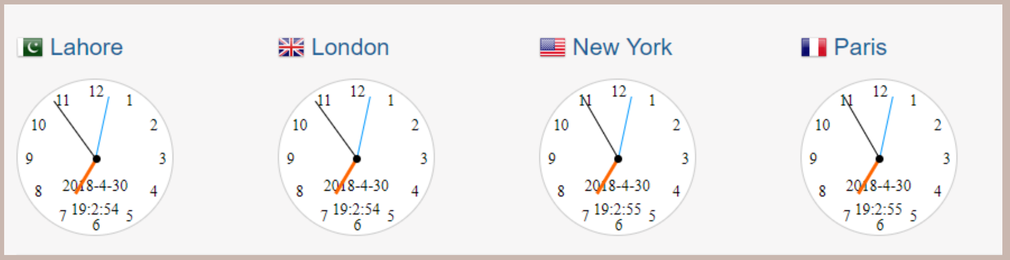
Javascript vs Java

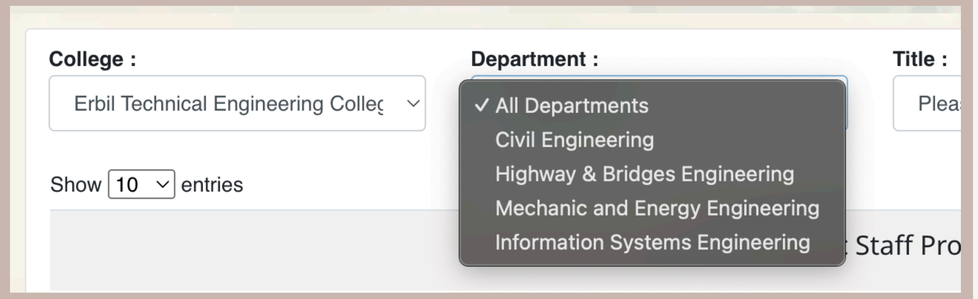
Javascript vs Java

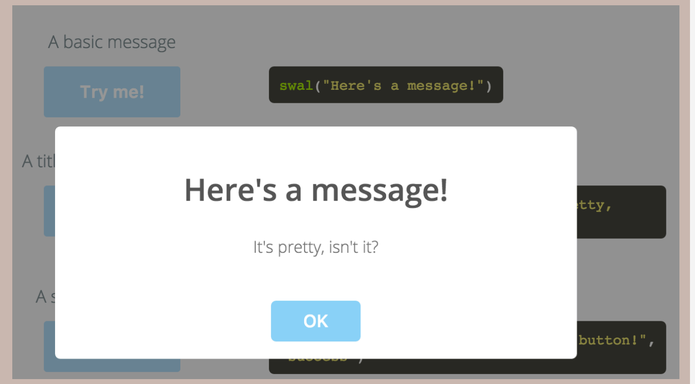
- Interpreted, Not Compiled: JavaScript code is executed line by line in the browser or a runtime environment, making it interpreted rather than compiled.
- More Relaxed Syntax and Rules: JavaScript has a more lenient syntax compared to Java. It allows implicit type conversions and dynamic typing.
  - Fewer and "looser" data types.
  - Variables don't need to be declared (not recommended, still working)
  - Errors Often Silent (Few Exceptions): JavaScript often fails silently in the case of minor errors, unlike Java, where strict error handling is enforced.
- Contained within a web page and integrates with its HTML/CSS content

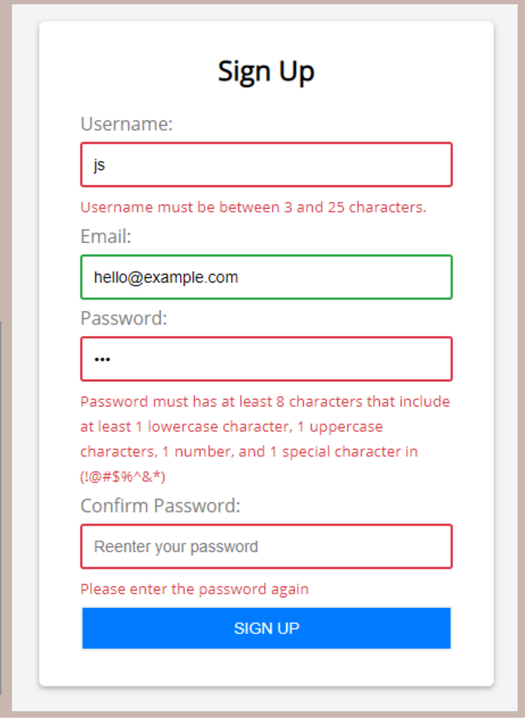
Application of JavaScript

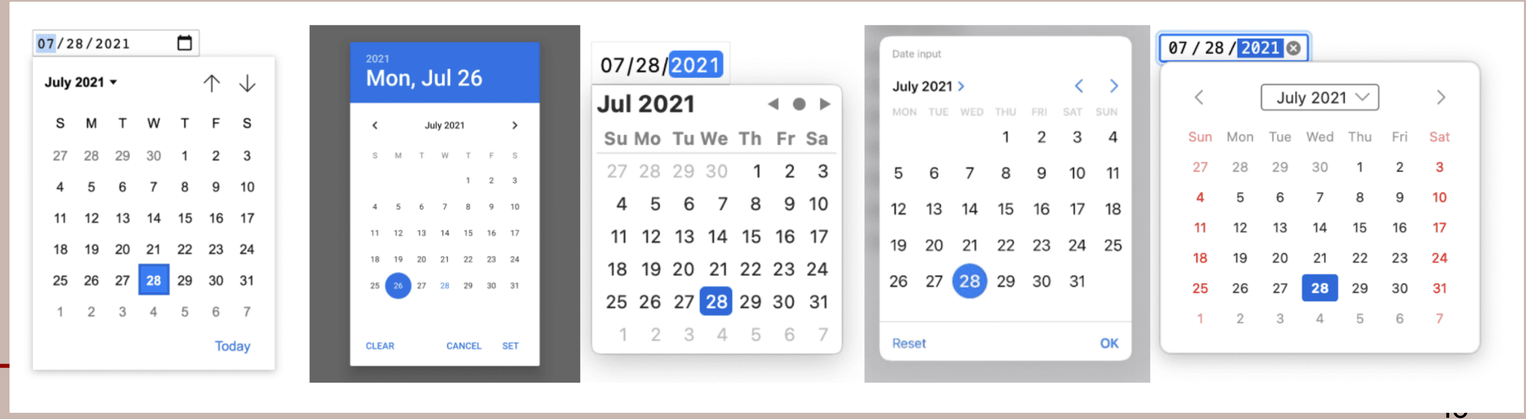
- Client-side validation,
- Dynamic drop-down menus,
- Displaying date and time,
- Displaying pop-up windows and dialog boxes
- Displaying clocks











Code Example

```
<script>
document.write("Hello JavaScript by JavaScript");
</script>
```

```
<script type="text/javascript">
document.write("JavaScript is a simple language for javatpoint learners");
</script>
```

3 Places to put JavaScript code

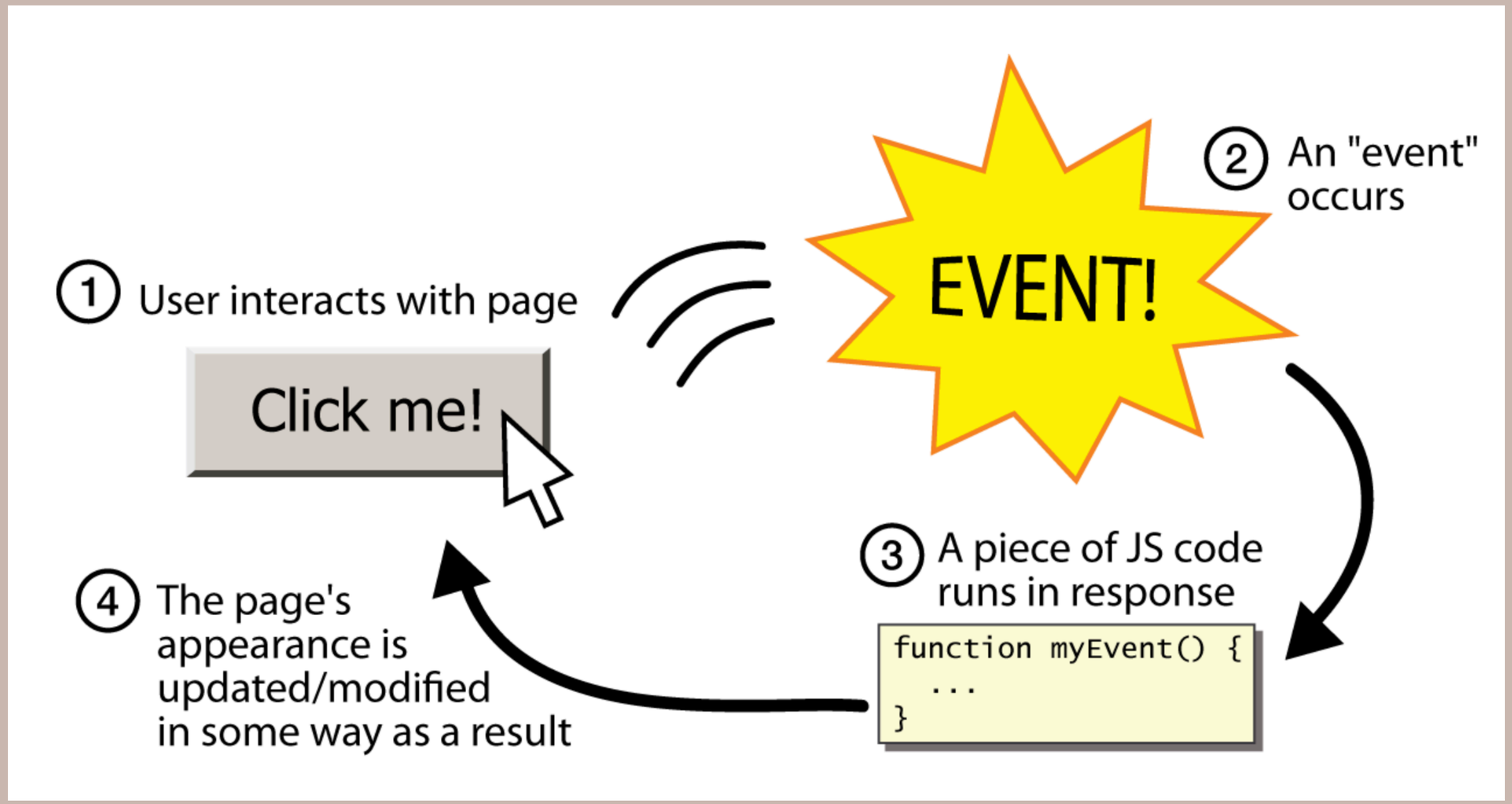
- Between the body tag of html
- Between the head tag of html
- In .js file (external javaScript)

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript in Body</title>
</head>
<body>
    <script>
        document.write("Hello Kurdistan!");
    </script>
</body>
</html>
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript in Body</title>
    <script type="text/javascript" src="script.js"></script>
</head>

<body>



</body>
</html>
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript in Body</title>
    <script>
            alert("Hello, Kurdistan!");
    </script>
</head>
<body>

</body>
</html>
```

Event-driven programming



Practical Examples

Advantages of External JavaScript

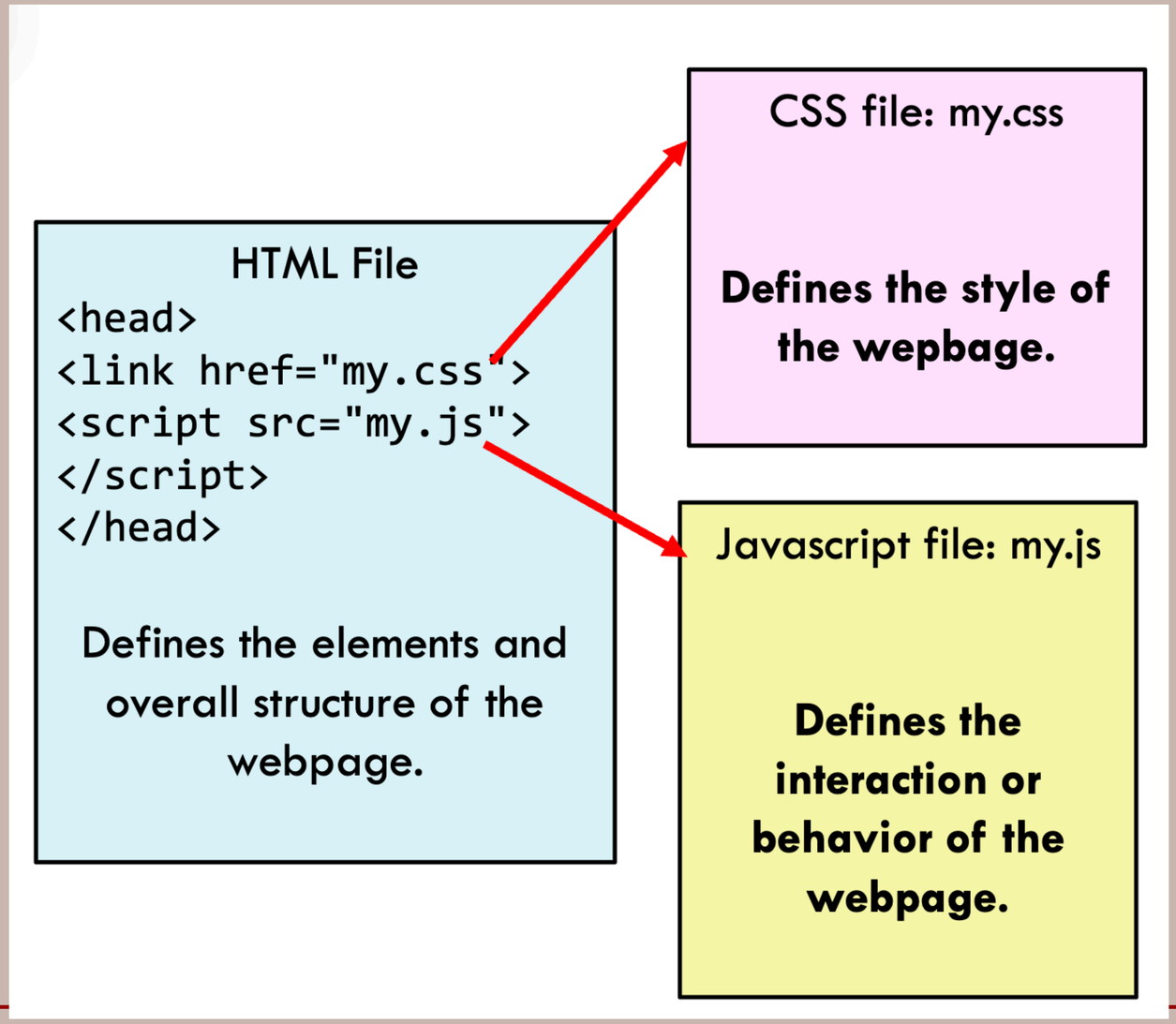
- It helps in the reusability of code in more than one HTML file.
- It allows easy code readability.
- It is time-efficient as web browsers cache the external js files, which further reduces the page loading time.
- It enables both web designers and coders to work with html and js files parallelly and separately.
- The length of the code reduces as only we need to specify the location of the js file.

JavaScript Comment

```
<script>
    // It is single line comment
    document.write("hello javascript");
</script>

<script>
    var a=10;
    var b=20;
    var c=a+b; //It adds values of a and b variable
    document.write(c); //prints sum of 10 and 20
</script>

<script>
    /* It is multi line comment.
    It will not be displayed */
    document.write("example of javascript multiline comment");
</script>
```

HTML + CSS + JavaScript

HTML + CSS + JavaScript



Variables and rules

```
<script>
    var clientName = "Connie Client"; /* variable clientName */
    var age = 32; // variable age
    var id = 3994330; // variable id
</script>
```

- First character must be a letter or an underscore (_)
- The rest of the variable can be any letter, number, or underscore
- Variable names are case sensitive
- age, Age, AGE would all be different variable names
- You cannot use JavaScript reserved words for variable names
- Example of a reserved word?
  - var var var = 10; // Since var is used to declare a variable, you can't use **var** as a variable

Variable name examples

- Valid names:

```
_myVar      thissisalongvariablename      num
_var        eecs1012                      myString
name1       test_1                        x
```

- Invalid names:

```
1test       /* starts with a number */
test 1      /* there is a space in the name */
t$est       /* non alphanumeric character */
var         /* reserved word */
```

JS data types

TYPE	Explanation	Example
`Number`	Integers and numbers with decimal places.	99, 2.8, 5, -10
`String`	A variable that is a collection of characters. Sometimes we call this a string literal.	“Hello”, “EECS1012”
`Boolean`	A variable that wholes only two possible values – **true** or **false**.	true or false
`Array`	A variable that is actually a collection of variables that can be access with an index	[1,2,3,4, …] [“hello”, “deaner”, …]
`Objects`	Objects are special data types that have functions and data associated with them. These are more common in JS than PHP and we will need to use them often.	Document.getElementByID(); (example of an object)

`function`	A user defined function that can be called by an user event (e.g. mouse click, etc)	`function name ()`
		`{`
		`statements;`
		`..`
		`}`

Variables

```
let x = 42;              // x is a number
x = "Hello";             // now x is a string
x = true;                // now x is a boolean
```

Arithmetic Operators

```
<script>

    var x = 10;
    let y = 20;
    const z = 30;
    a = 23;

    var result = x+y-(3*z)/a
    document.write(result); // 26.08695652173913

</script>
```

What are the differences among all four variables?

“Short hand” assignment operators

Assignment	Same as:	
`a += b;`	`a = a + b;`	Addition
`a -= b;`	`a = a - b;`	Subtraction
`a *= b;`	`a = a * b;`	Multiplication
`a /= b;`	`a = a / b;`	Division
`a %= b;`	`a = a % b;`	Modulus
`a++;`	`a = a + 1;`	Self Addition
`a--;`	`a = a -1;`	Self subtraction

String variables

```
<script>

    var x = "Hello ";
    var y = "World";
    var z = x + y;
    document.write(z); // s3 = "Hello World"

    var length = z.length;
    document.write("<br> length = " + length); // length = 11

</script>
```

Arrays

What are the differences?

```
<script>

    var names = ["Aram", "Kardo", "Kamal" ];
    // We can access each individual value using the following notation.
    console.log(names[0]);
    document.write(names[1]);

    var size = names.length; // This will return the size of the array.

    document.write("<br>");
    document.write(size);

</script>
```

Arrays can mix datatypes

```
var car = ["Saab", "Volvo", "BMW" ];    // Array of Strings

var nums = [ 1, 2, 3, 4, 5 ];           // Array of Numbers

var data = ["EECS1012", 780, "Fall", 2018 ];  // Mix types
```

Control Statements (if Statement)

```
<script>

    let age = 18;

    if (age < 18) {
        document.write("Minor");
    } else if (age == 18) {
        document.write("Just became an adult");
    } else {
        document.write("Adult");
    }


</script>
```

Control Statements (switch)

```
<script>

    let grade = 'A';

    switch (grade) {
        case 'A':
            console.log("Excellent");
            break;
        case 'B':
            console.log("Good");
            break;
        default:
            console.log("Needs Improvement");
            break;
    }


</script>
```

Control Statements (for-loop)

```
<script>
    // First loop: Iterates 5 times and prints "Iteration" followed by the current value of 'i'
    for (let i = 0; i < 5; i++) {
        document.write("Iteration " + i);
    }

    var name = "Tishk International University";

    // Adding a line break
    document.write("<br>");

    // Second loop: Iterates through each character in the 'name' string and prints it on a new line
    for (let i = 0; i < name.length; i++) {
        document.write(name[i] + "<br>");
    }

    document.write("<br>");

    // Declaring an array 'myArray' with different fruit names
    var myArray = ["Apple", "Banana", "Orange", "Mango", "Pineapple"];

    // Third loop: Iterates through each element of the 'myArray' array and prints it on a new line
    for (let i = 0; i < myArray.length; i++) {
        document.write(myArray[i] + "<br>");
    }
</script>
```

Control Statements

while & do-while loops?

Check it

JavaScript Functions

1)

```
function name() {
 statements;
  …
}
```

Keyword function is used to define a function.
name is the name of the function.
The parenthesis are used to denote it is a function that accepts no parameters.

2)

```
function name(parameter1, parameter2, …) {
 statements;
  …
}
```

This syntax allows parameters to be "passed" to the function.
Parameter names are defined between the (..)

3)

```
function name(parameter1, parameter2, …) {
 statements;
  …
 return value;
}
```

This syntax allows parameters to be "passed" to the function.

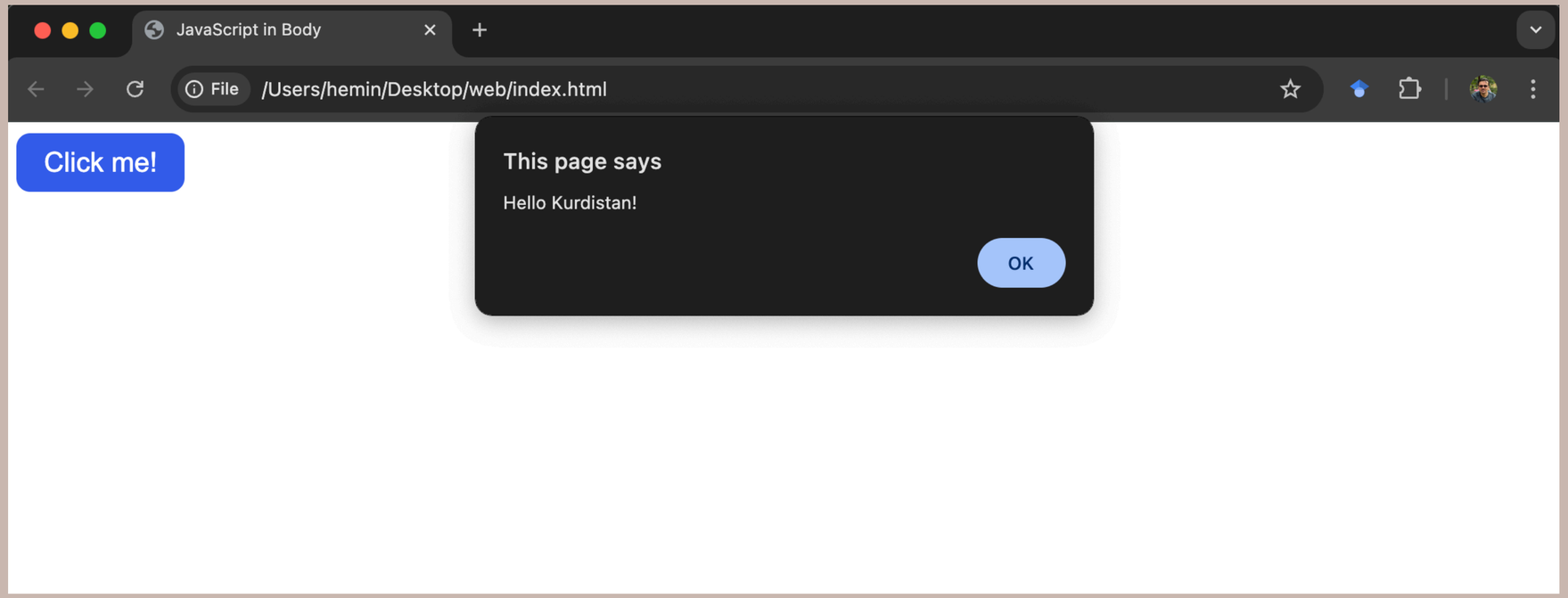
Note that the function **also** returns a value.

JavaScript Functions

```
<body>

    <button onclick="myFunction();">Click me!</button>


    <script>
        function myFunction() {
            alert("Hello Kurdistan!");
        }
    </script>

</body>
```

JavaScript Functions

```html
<body>

    <button onclick="myFunction();">Click me!</button>


    <script>
        function myFunction() {
            alert("Hello Kurdistan!");
        }
    </script>

</body>
```



JavaScript Objects

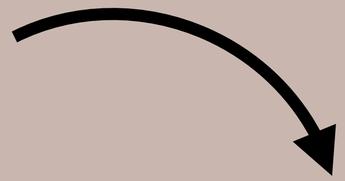
- Number and string variables are containers for data
- Objects are similar, but can contain many values.
- Objects also can associated functions (they are called methods to distinguish them from the functions you just learned about).

Math Object

```
/*  PI is value associated with the Math object.  We access it
using the "." operator, just like we did with length for arrays
and strings.  num now equals 3.141592653589793                    */
var num1 = Math.PI;

var num2 = -50.30;
var num3 = 4;
var num4 = 66.84

var result1 = Math.round(4.7);   // method rounds a number
var result2 = Math.abs( num2 ) ; // method computes absolute value
var result3 = Math.sqrt( num3 ); // method computes the square root
var result4 = Math.min( num2, num3 ); // returns the minimum of a list of nums
var result5 = Math.max(num2, num3);  // returns the maximum of list of nums
var result6 = Math.floor(num4);  // rounds number down to nearest integer
var result7 = Math.ceil(num4);   // rounds up to nearest integer
```

Date object

```
<script>
    var myDate = new Date();
    var year = myDate.getFullYear(); // returns the year
    var month = myDate.getMonth(); // returns the month
    var minute = myDate.getMinutes(); // returns the minute
    var second = myDate.getSeconds(); // returns the seconds
    var dateStr = myDate.toDateString(); // returns a string of the date

    document.write("Day: " + myDate + "<br>");
    document.write("Year: " + year + "<br>");
    document.write("Month: " + month + "<br>");
    document.write("Minute: " + minute + "<br>");
    document.write("Second: " + second + "<br>");
    document.write("Date String: " + dateStr + "<br>");

</script>
```



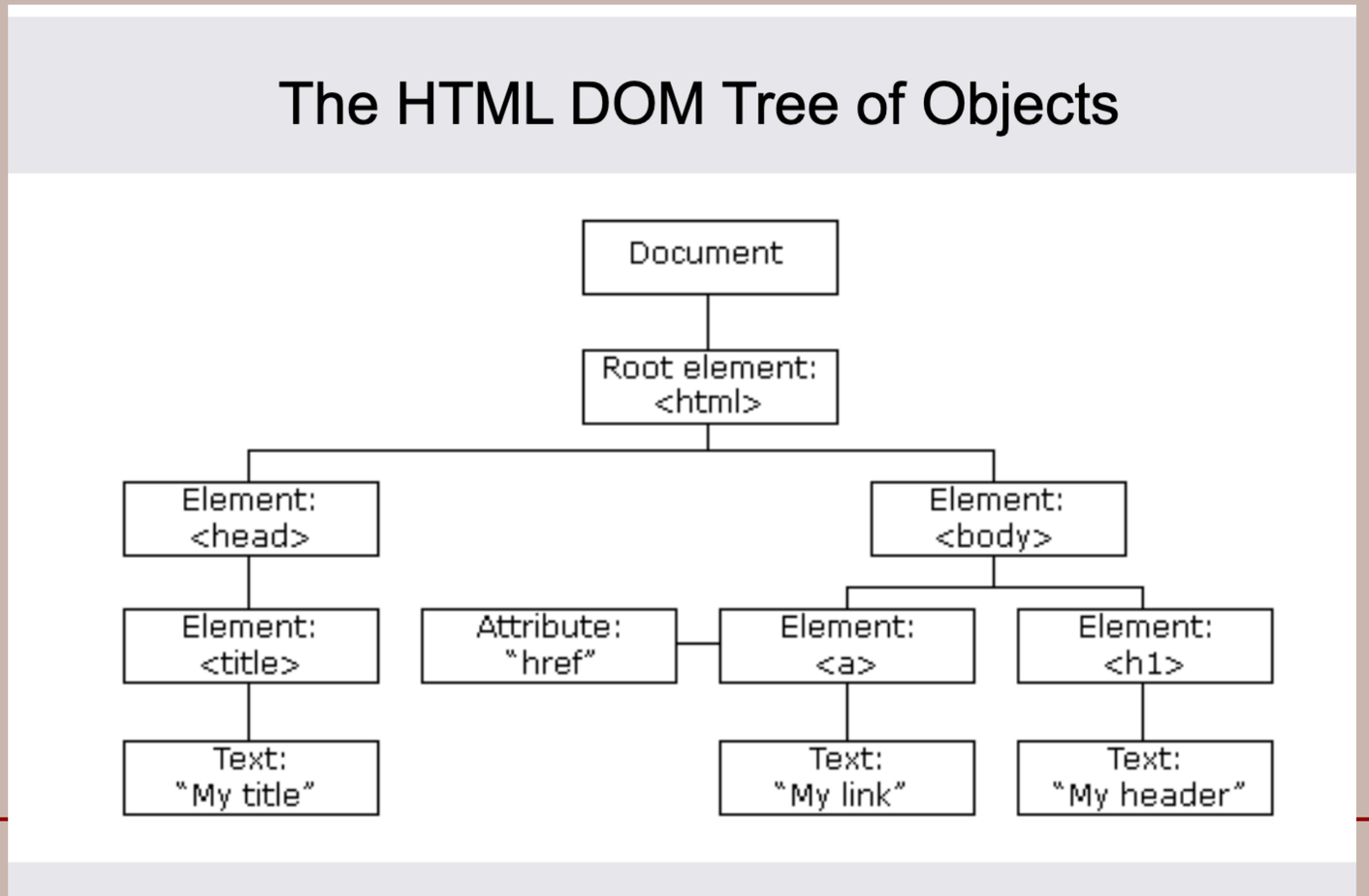
Day: Tue Oct 22 2024 07:04:49 GMT+0300 (Arabian Standard Time)
Year: 2024
Month: 9
Minute: 4
Second: 49
Date String: Tue Oct 22 2024

Examples - Output?

```
<script>

    function variableDemo(x) {
        x = x * 2;
        return x;
    }

    let result = variableDemo(5);

    document.write(result); // What is this output?

</script>
```

The Document Object Model (DOM)

- The DOM is a programming interface for HTML documents.
- Represents the page so programs can change the document structure, style, and content.



document Object

```
document.getElementById("id name")
```

Object "document".
Note that the object name is **lowercase**!

Call the object's method getElementById(. . .) searches the HTML page to find the element with the id=="id name".

document Object

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript in Body</title>
</head>

<body>

    <p id="uni"></p>
    <script>
        document.getElementById("uni").innerHTML = "Welcime to my Website!";
    </script>

</body>
</html>
```

document Object

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript in Body</title>
</head>

<body>

    <p id="uni"></p>
    <script>
        document.getElementById("uni").innerHTML = "Welcime to my Website!";
    </script>

</body>
</html>
```

- **document.getElementById("uni"):** This function selects the HTML element with the id attribute set to “uni".
- **.innerHTML**: This property allows you to change or update the HTML content inside the selected element.

Example

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Change HTML Content</title>
</head>
<body>

    <!-- HTML element with id "uni" -->
    <h1 id="uni">Original Content</h1>

    <!-- Button to trigger the change -->
    <button onclick="changeContent()">Change Content</button>

    <!-- JavaScript to change content -->
    <script>
        function changeContent() {
            // Selects the element with id "uni" and changes its content
            document.getElementById("uni").innerHTML = "Welcome to my Website!";
        }
    </script>

</body>
</html>
```

Example

```
<!DOCTYPE html>
<html>
    <head>
        <script>
        function validateForm() {
            let x = document.forms["myForm"]["fname"].value;
            if (x == "") {
                alert("Name must be filled out");
                return false;
            }
        }
        </script>
    </head>
    <body>

        <h2>JavaScript Validation</h2>

        <form name="myForm" action="#" onsubmit="return validateForm()" method="post">
        Name: <input type="text" name="fname">
        <button type="submit">Submit</button>
        </form>

    </body>
</html>
```

Example

```
<!DOCTYPE html>
<html>
    <head>

    </head>
    <body>

        <p id="text">This is a paragraph.</p>
        <button onclick="changeColor()">Change Color</button>

        <script>
            function changeColor() {
                document.getElementById("text").style.color = "red";
            }
        </script>

</body>
</html>
```

Example

```
<div id="container"></div>
<button onclick="addParagraph()">Add Paragraph</button>

<script>
    function addParagraph() {
        // Create a new paragraph element
        let newPara = document.createElement("p");
        newPara.innerHTML = "I am a new paragraph!";

        // Append the new paragraph to the container div
        document.getElementById("container").appendChild(newPara);
    }
</script>
```

Example

```html
<input type="text" id="nameInput" placeholder="Enter your name">
<button onclick="showName()">Submit</button>
<p id="display"></p>

<script>
    function showName() {
        // Get the value of the input field
        let name = document.getElementById("nameInput").value;

        // Display the name in the paragraph
        document.getElementById("display").innerHTML = "Hello, " + name + "!";
    }
</script>
```

Thank You