Disclaimer

The information and content provided in this presentation are for educational purposes only. Delta Tech Info and its affiliates do not endorse or promote any specific products or services mentioned in this presentation. The viewers are advised to exercise their own judgment and discretion while applying the knowledge gained from this presentation.

Delta Tech Info

- Delta Tech Info is a platform dedicated to spreading digital literacy among the Pakistani youth. Our aim is to equip individuals with the knowledge and skills necessary to navigate the digital world effectively. Through our courses and educational resources, we strive to empower the youth to harness the power of technology and enhance their opportunities in today's digital age."

What is a programming language?

- A **programming language** is a computer language programmers use to develop software programs, scripts, or other sets of instructions for computers to execute.

What is JavaScript?

- JavaScript is one of the **3 languages** all web developers **must** learn:
- 1. **HTML** to define the content of web pages
- 2. **CSS** to specify the layout of web pages
- 3. **JavaScript** to program the behavior of web pages
- JavaScript is the world's most popular programming language.
- JavaScript is a lightweight programming language that web developers commonly use to create more dynamic interactions when developing web pages, applications, servers, and or even games.

Using JS in html document

- There are two ways to add JS to your html doc.

1. Using <script> tag
   ex:
   <script> …..Js code will be here…. </script
2. Using external .js file. (This is prefrable makes code easy to read)
   <script src="myScript.js"></script>

Output in JS

- Output in JS is of 4 types

1. console.log(): through this we can print msg in console
2. alert(): simple alert window appears in browser window
3. window.innerHTML : writing into html element
4. Window.write(): overwrites whole of document (for testing purpose only)

Statements syntax

- JavaScript statements are composed of:
- Values, Operators, Expressions, Keywords, and Comments.
- The statements are executed, one by one, in the same order as they are written.
- JS statements can be separated by just keeping eacht statement on different line
- Semicolons separate JavaScript statements.
- Add a semicolon at the end of each executable statement: This helps getting more than one statements on one line

  EX:

  ```
  let a = 5
  let b = 6;
  let c = a + b; a=6;
  // this is how to use comment in js
  ```

Variables

- **Variables are used to store information to be referenced and manipulated in a computer program**. They also provide a way of labeling data with a descriptive name
- Three ways to declare variables in JS
- Var (depreciated)
- Let these are values that are mutable(value can be changed after assigning for the first time.
  EX:
  ```
  let a = "fahad";
  alert(a);
  a = "Ali"
  alert(a)
  ```
- Const keyword declares variable that are immutable (cannot be changed) only assigned value for the first time.
  ```
  Const a = "fahad"
  alert(a);
  a= "Ali" // will give and error
  ```

Operators in JS

Operators in JavaScript

Operators are symbols or keywords that perform operations on values or variables.

Arithmetic Operators

Arithmetic operators are used to perform mathematical calculations.

Examples:

- Addition: **+**
- Subtraction: **-**
- Multiplication: *****
- Division: **/**
- Modulo: **%**

Assignment Operators

Assignment operators are used to assign values to variables.

Examples:

- Assignment: **=**
- Addition assignment: **+=**
- Subtraction assignment: **-=**
- Multiplication assignment: ***=**
- Division assignment: **/=**
- Modulo assignment: **%=**

Comparison Operators

Comparison operators are used to compare values and return a Boolean result.

Examples:

- Equal to: **==**
- Not equal to: **!=**
- Greater than: **>**
- Less than: **<**
- Greater than or equal to: **>=**
- Less than or equal to: **<=**

Operators in JS (Cont)

Logical Operators

Logical operators are used to combine or negate conditions.

Examples:

- Logical AND: **&&**
- Logical OR: **||**
- Logical NOT: **!**

Increment and Decrement Operators

Increment and decrement operators are used to increase or decrease the value of a variable.

Examples:

- Increment: **++**
- Decrement: **--**

String Operators

String operators are used for concatenating strings.

Examples:

- Concatenation: **+**

Conditional (Ternary) Operator

The conditional operator provides a shorthand way of writing if-else statements.

Example:

- **condition ? expression1 : expression2**

Datatype in JS

String

- Represents a sequence of characters.
- Example: "Hello, World!"

Number

- Represents numeric values, including integers and floating-point numbers.
- Example: 42

BigInt

- Represents arbitrary precision integers.
- Example: 9007199254740991n

Boolean

- Represents a logical value: true or false.
- Example: true

Datatypes in JS (cont)

Undefined

- Represents a variable that has been declared but not assigned a value.
- Example: let x;

Null

- Represents the intentional absence of any object value.
- Example: let y = null;

Symbol

- Represents a unique identifier, often used as object properties.
- Example: const id = Symbol("unique");

Object

- Represents a collection of key-value pairs or complex data structures.
- Example: const person = { name: "John", age: 30 };

Object Datatypes in JS

Object

Represents a collection of key-value pairs or properties.

Objects can store and organize related data and functionality.

Array

Represents an ordered list of values.

Arrays can store multiple values in a single variable and provide methods for manipulation.

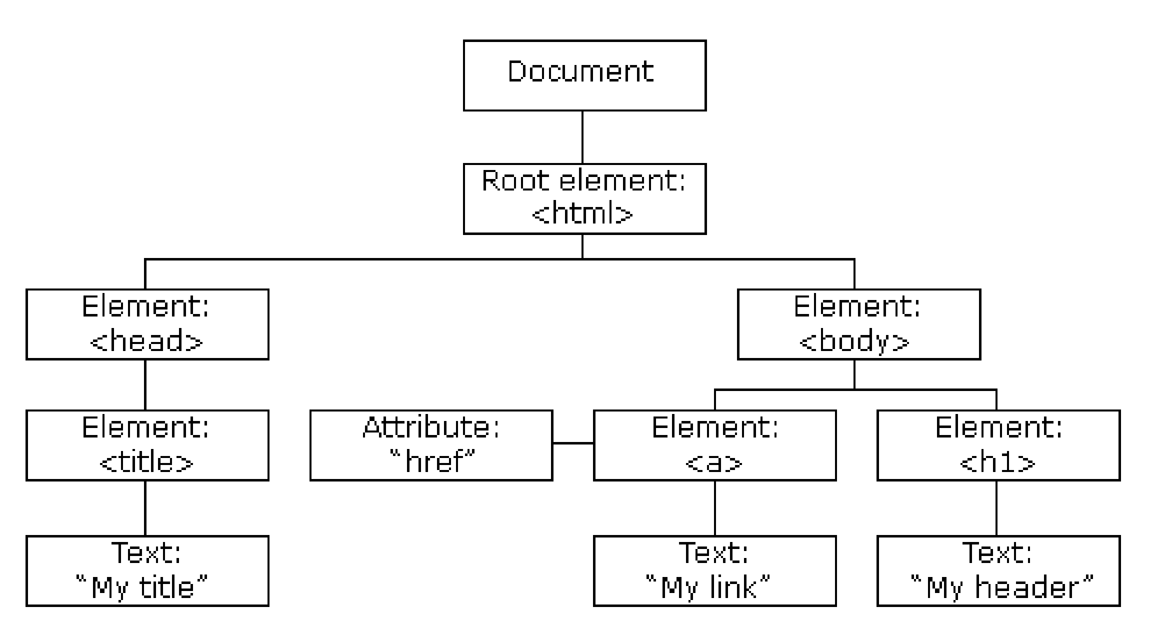
Date

Represents a specific point in time or a combination of date and time.

Date objects provide methods for working with dates, times, and timezones.

Introducing DOM

- With the HTML DOM, JavaScript can access and change all the elements of an HTML document.
- The HTML DOM (Document Object Model)
- When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel of the page.
- The **HTML DOM** model is constructed as a tree of **Objects**:



DOM Capabilities

Model	With the object model, JavaScript gets all the power it needs to create dynamic HTML:
Change	JavaScript can change all the HTML elements in the page
Change	JavaScript can change all the HTML attributes in the page
Change	JavaScript can change all the CSS styles in the page
Remove	JavaScript can remove existing HTML elements and attributes
Add	JavaScript can add new HTML elements and attributes
React	JavaScript can react to all existing HTML events in the page
Create	JavaScript can create new HTML events in the page

Example of using DOM

- <!DOCTYPE html>
- <html>
- <head>
- <title>innerHTML Example</title>
- </head>
- <body>
- <div id="myElement">Hello, World!</div>
- <button onclick="changeText()">Change Text</button>
-
- <script src="script.js">
  function changeText(){
- let element = document.getElementById("myElement");
- element.innerHTML = "New Text!";

}

Thank You!!!