**Web Designing Assignment**

**Term-4**

**Module(Javascript Basic Amp; & Dom)**

1. What is JavaScript?

* JavaScript is a high-level, interpreted programing language primarily used for making web pages interactive. It enables dynamic content update, animation, form validations and communication with web servers.

1. What is the use of isNaN function?

* In JavaScript NaN is short for “Not-a-Number”. The isNaN() method returns true if a value is NaN. The isNaN method converts the value to a number before testing it.

1. What is negative infinity?

* Negative infinity is a special numeric value that is returned when an arithmetic operation or mathmatical function generates a negative value greater than the largest representable number in javascript.

1. Which company developed Javascript?

* JavaScript was developed by Netscape Communications Corporation. It was created in 1995 by Brendan Eich, while he was working at Netscape. The language was originally

called Mocha, then renamed to LiveScript, and finally to JavaScript.

1. What are undeclared and undefined variables?

* Undeclared Variables

A Variables that has not been declared at all in your code before you try to use it.

This usually results in a ReferenceError.

* Undefined Variables

A Variables that has been declared but has not been assigned a value yet.

Its value is automatically set to undefined.

1. Write the code for adding new elements dynamically?

* Sure! Here’s a simple example using JavaScript to dynamically add new elements (like a new <div>, <li>, etc.) to the Dom.

<!DOCTYPE html>

<html>

<head>

<title>Dynamic Element Addition</title>

</head>

<body>

<h2>Dynamic List</h2>

<ul id="myList">

<li>Item 1</li>

<li>Item 2</li>

</ul>

<button onclick="addListItem()">Add Item</button>

<script>

let itemCount = 2;

function addListItem() {

itemCount++;

// Create a new list item

const li = document.createElement("li");

li.textContent = "Item " + itemCount;

// Append it to the list

document.getElementById("myList").appendChild(li);}

</script>

</body>

</html>

1. What is the difference between ViewState and SessionState?

* In Javascript (especially when working with web applications), ViewState and SessionState aren’t native Javascript concepts, but they are often discussed in the context of ASP.NET or server-side web development.
* ViewState

Scope:- Page-level (specific to a single page).

Lifecycle:- Maintained during the postbacks of the same page.

Storage:- Stored in a hidden field on the page itself (on the client side).

Purpose:- Preserves values of page controls between postbacks.

Size limitations:- Can bloat the page because it is sent back and forth with each request/response.

Security:- Vulnerable to tampering unless encrypted or hashed.

* SessionState

Scope:- User-session level (shared across all pages during the session ).

Lifecycle:- Maintained throughout the user’s session(usually util timeout or logout).

Storege:- Stored on the server.

Purpose:- Stores user-specific data like login info, preferences, shopping cart etc.

Size Limitations:- Less limited, since it’s stored server-side.

Security:- More Secure (data isn’t exposed the client).

1. What is === operator?

* The === operator in JavaScript is the strict equality operator. It checks whether two values are equal in both value and type.
* === does not perfrom type coercion. It only return true if both the value and tyape match.

1. How can the style/class of an element be changed?

* You can change the style or class of an element in javascript using a few different approaches, depending on what you want to do
* Change the class

Ex. document.getElementById("myElement").className = "new-class";

Ex. document.getElementById("myElement").classList.add("my-class");

1. How to read and write a file using JavaScript?

* Reading and writing files using JavaScript depends on where you’re running your JavaScript
* Web browsers do not allow direct file system access for security reasons.But you can:

Read files using <input type=”file”>and the fileReader API

Trigger downloads (write) by creating and downloading a Blob

1. What are all the looping structures in JavaScript?

* Loops can execute a block of code a number of times.
* Loops are handy, if you want to run the same code over and over again, each time with a different value.
* For Loop :- Executes a block of code a specific number of times.

For (let I = 0; I < 5; i++) {

Console.log(i)

}

* While Loop :- Runs as long as the condition is true.

let i = 0;

while (i < 5){

console.log(i);

i++;}

* Do While Loop :- Executes the block at least once, the continues while the condition is the true

Let i = 0;

do {

console.log (i);

i++;

}While (i < 5);

* For in Loop :- lterates over the enumerable properties of an object (inludig inherited once).

const person = { name: “Alice”, age : 30 };

for (let key in person) {

console.log(key, person [key]);

}

* For of Loop :- lterates over iterable object like arrays, string, maps, sets, etc.

const arr = [10, 20, 30];

for (let value of arr) {

console.log(value);

}

1. How can you convert the string of any base to an integer in JavaScript?

* In JavaScript, you can convert a string from any base (between 2 and 36) to an integer using the built-in parseInt() function. Here’s the syntax:

parseInt(string, radix);

* String : The value you want to parse.
* Radix : The base of the numeral system (between 2 and 36)

// Binary (base 2)

console.log(parseInt("1010", 2)); // Output: 10

// Octal (base 8)

console.log(parseInt("17", 8)); // Output: 15

// Decimal (base 10)

console.log(parseInt("123", 10)); // Output: 123

// Hexadecimal (base 16)

console.log(parseInt("1f", 16)); // Output: 31

* Always provide the radix to avoid unexpected result (especially with strings that start with 0).
* If the string Contains invalid characters for the given base, parsing stops at the first invalid character.

1. What is the function of delete operator?

* The delete operator in javaScript is used to remove a property from a object.

var person = {

firstname :- “spicy”,

lastname :- “sins”,

age :- 50,

eyeColor :- “brown”

};

Delete person.age

* The delete operator deletes both the value of the property and the property itselt.
* After deletion, the property cannot be used before it is added back again.
* The delete operator is designed to be used on object properties. It has no effect on variables or functions.

1. What are all the types of Pop up boxes available in JavaScript?

* JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box,
* Alert Box

An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click “OK” to proceed.

* Example

Alert("This is an alert box!”);

* Confirm Box

A confirm box is often used if you want the user to verify or accept something.

When a confirm box pops up, the user will have to click either “OK” or “CANCEL” to proceed.

If the user clicks “OK”, the box returns true. If the user clicks “cancel”, the box return false.

* Example

if (confirm("Press a button!")) {  
  txt = "You pressed OK!";  
} else {  
  txt = "You pressed Cancel!";  
}

* Prompt Box

A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either “OK” or “Cancel” to proceed after entering an input value.

If the user clicks “OK” the box returns the input value. If the user clicks “Cancel” the box returns null.

* Example

let person = prompt("Please enter your name", "Harry Potter");  
let text;  
if (person == null || person == "") {  
  text = "User cancelled the prompt.";  
} else {  
  text = "Hello " + person + "! How are you today?";  
}

* Line Breaks

To display line breaks inside a popup box, use a back-slash followed by the character n.

* Example

Alert("Hello\nHow Are you?”)

1. What is the use of Void(0)?

* Void(0) is a JavaScript expression often used in links (<a> tags) to prevent the browser from performing its default action, which is usually navigating to another page.
* The void operator evaluates an expression and returns undefined.
* So, void(0) evaluates the number 0 and returns undefined.
* When used in an herf, like this:

Ex:- <a href=”javascript:void(0)”>Click me</a>

1. How can a page be forced to load another page in JavaScript?

* In JavaScript, you can force a page to load another page by changing the window.location or using other methods like window.location.assign() or window.location.replace().
* Here are a few common ways:

1. Window.location.href

window.location.href = ‘https://example.com’;

* behaves like clicking a link.
  + Adds the new page to browser history(so back button works).

1. Window.location.assign()

window.location.assign(‘https://example.cpm’);

* + Similar to href, also keeps the current page in history.

1. Window.location.replace()

window.location.replace(‘https://example.com’);

* + Replaces the current page in browser history.
  + Back button won’t return to previous page.

1. What are the disadvantages of using innerHTML in JavaScript?

* Using innerHTML in JavaScript in convenient, but it comes with several disadvantages that can impact security, performance, and maintainability.

1. Security Risks(XSS Vulnerabilities)

If you insert user-generated content using innerHTML, you risk exposing your site to cross-site (XSS) attacks.

1. Performance Issues

When you use innerHTML, the browser parses and re-renders the entire inner content of the element ---even ef you’re just changing a small part.

This can lead to unnecessary reflows and repaints, slowing down your page.

1. Loss of Event Listeners

Any DOM elements inside the target element that had event listeners attached will lose them after using innerHTML.

This happens because innerHTML replaces the whole DOM tree, not just the text.

1. No Validation or Sanitation

innerHTML accepts any valid HTML, even broken or malformed code, which can lead to unpredictable behavior or bugs.

1. Harder to Maintain

Mixing HTML and JavaScript logic through innerHTML can make your code harder to read, debug, and maintain especially in larger projects.