

## Lab Session 3: Initiation to JavaScript

### Introduction

JavaScript is used to make HTML websites interactive. It can be embedded directly inside an HTML document using:

```
<script>  
// JavaScript code here  
</script>
```

Or called from an external file: `<script type="text/javascript" src="example.js"></script>`

### General Rules

- Every statement ends with a semicolon ';'.  
• Decimal numbers use a dot (.), not a comma (,).
- JavaScript is case-sensitive.
- Comments: - Single-line: // comment - Multi-line: /\* comment \*/
- JavaScript is object-oriented: objects have methods, properties, and can contain other objects (e.g., window, document, or custom objects).
- Control structures are similar to Java and C: if, else, for, while, switch, etc.
- Variable declaration:
  - Use var, let, or const - Example: var x = 10;
  - JavaScript is dynamically typed
  - Common types: boolean, number, string, function, object
  - Use typeof to check a variable's type, e.g., typeof(3.14) returns 'number'

### Basic Exercises

#### Exercise 1: Alerts, Confirms, and Prompts

Create a webpage with a button. On click, display the following:

- alert("Welcome!")
- confirm("Do you want to continue?")
- prompt("Enter your name:")

Use document.write("Hello " + name) to show the result on the page.

## DOM Manipulation

### Exercise 2: innerHTML

Create a paragraph element like this:

```
<p id="pid">Some text</p>
```

JavaScript:

```
let x = document.getElementById("pid");  
alert(x.innerHTML);
```

```
x.innerHTML = "New text";
```

## Events and Styles

### Exercise 3: Change Style on Click

Create a <div> with some text. When clicked:

- Change the text color
- Change the font style

Then redo the same exercise by using an external JavaScript file.

## Dynamic Images

### Exercise 4: Change Images

1. Create a button that toggles between Image 1 and Image 2 every time it's clicked.
2. Extend it to switch between three images.
3. Add the following behaviors:
  - **onmouseover**: switch to Image 2
  - **onmouseout**: switch back to Image 1

## Control Structures and Input Handling

### Exercise 5: Numeric Input Validation

Create a text input field to validate numbers:

1. Option 1: Click a "Validate" button to check the entire input
2. Option 2: Validate character-by-character using event.keyCode

**Exercise 6: Password Validation with Limited Attempts**

1. Create a password input and a "Submit" button.
2. If the password is correct:
  - Show: "Your password is correct"
  - Load a new HTML page
3. If incorrect:
  - Show: "Incorrect password, try again"
  - Clear the input
4. Limit attempts to 5:
  - After 5 failed tries, show "No more attempts allowed" and close the page

**Form Validation****Exercise 7: Simple Form Validation**

Create a form with a text input.

- If the field is empty:
  - Show an error message next to the input
  - Prevent the form from being submitted
- If the field contains data:
  - Show: "Form can be submitted"

**Exercise 8: Live Field Validation**

As the user types in the input:

- If the input has fewer than 3 characters, show an error message in red
- If the input is valid, show a success message in green

**Additional (Advanced) Exercises****Exercise 9: Random Background Color**

Create a button. On click, change the background color of the page randomly.

**Exercise 10: Simple Calculator**

Create two input fields (a and b).

- Add a dropdown menu to choose an operation (+, -, \*, /)
- Display the result dynamically.

**Exercise 11: To-Do List**

Create an input field to enter a task.

- Display the list of tasks below.
- Allow deletion of tasks by clicking on them.

**Mouse Events:**

- **onclick**: Click on the element
- **ondblclick**: Double-click on the element
- **onmousedown**: Mouse button pressed down on the element
- **onmousemove**: Mouse moved over the element
- **onmouseover**: Mouse entered the element
- **onmouseout**: Mouse left the element
- **onmouseup**: Mouse button released over the element

**Keyboard Events:**

- **onkeydown**: Key pressed down
- **onkeypress**: Key pressed and released
- **onkeyup**: Key released

**Form Events:**

- **onblur**: Form element loses focus
- **onchange**: Form element content changes
- **onfocus**: Form element gains focus
- **onreset**: Form is reset
- **onselect**: Text selected in a form field
- **onsubmit**: Form is submitted