**Tutorial 6 - JS**

Repo link: <https://github.com/tinkerness/S7_Web_Programming/tree/main/tutorials/tut6>

Hosted link: <https://tinkerness.github.io/S7_Web_Programming/tutorials/tut6/index.html>

1. **Create an HTML button element that, when clicked, changes the background color of a <div> element with the ID "myDiv" to red.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Button Onclick Event</title>

<style>

\* {

text-align: center;

}

#myDiv {

height: 100px;

background-color: orange;

}

</style>

</head>

<body>

<div id="myDiv"><h1>Hello!</h1></div>

<br/><br/>

<button id="changeColourBtn" > Change Colour </button>

<script>

document.getElementById("changeColourBtn").addEventListener("click", changeColour)

function changeColour() {

document.getElementById("myDiv").style.backgroundColor="red";

}

</script>

</body>

</html>

1. **Develop a simple image gallery. Create a group of thumbnail images, and when a user clicks on a thumbnail, display the corresponding full-sized image in a larger container.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Thumbnail Images</title>

<style>

.thumbnail {

width: 100px;

height: 75px;

margin: 5px;

cursor: pointer;

}

.image-container {

display: none;

}

</style>

</head>

<body>

<img src="boat\_tn.gif" alt="Image 1" data-fullsize="boat.jpg" class="thumbnail" />

<img src="flower\_tn.png" data-fullsize="flower.png" alt="Image 2" class="thumbnail" />

<img src="boat\_tn.gif" data-fullsize="boat.jpg" alt="Image 3" class="thumbnail" />

<div class="image-container">

<img src="" id="fullImage" alt="Full-sized Image" />

</div>

<script>

// tn-img

const thumbnails = document.querySelectorAll(".thumbnail");

// full-img

const imageContainer = document.querySelector(".image-container");

const fullImage = document.getElementById("fullImage");

// event listener

thumbnails.forEach(thumbnail => {

thumbnail.addEventListener("click", ViewFullImage);

});

function ViewFullImage() {

const fullSizeSrc = this.getAttribute("data-fullsize");

fullImage.src = fullSizeSrc;

imageContainer.style.display = "block";

}

</script>

</body>

</html>

1. **Implement a counter that increases by one every time a button is clicked. Display the current count on the web page.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Counter</title>

</head>

<body>

<p>Count: <span id="countDisplay">0</span></p>

<button id="incrementButton">Increase Count</button>

<script>

const countElement = document.getElementById("countDisplay");

let count = 0;

document.getElementById("incrementButton").addEventListener("click", function () {

count++;

countElement.textContent = count;

});

</script>

</body>

</html>

1. **Build a to-do list where users can add and remove tasks. Implement event handlers for adding tasks and deleting tasks.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>TO-DO List</title>

<style>

\* {

text-align: center;

}

ul {

list-style: none;

}

li {

display: flex;

justify-content: space-around;

align-items: center;

}

button {

margin: 5px;

cursor: pointer;

}

</style>

</head>

<body>

<h2>To-Do List</h2>

<input type="text" id="taskInput" placeholder="Enter a new task">

<button id="addTask" onclick="addTask()">Add Task</button>

<ul id="todoList">

<!-- To-Do list items -->

</ul>

<script>

function addTask() {

const taskText = document.getElementById("taskInput").value;

if (taskText) {

const taskList = document.getElementById("todoList");

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

listItem.textContent = taskText;

const removeButton = document.createElement("button");

removeButton.textContent = "Remove";

removeButton.onclick = function() {

taskList.removeChild(listItem);

};

listItem.appendChild(removeButton);

taskList.appendChild(listItem);

document.getElementById("taskInput").value = "";

}

}

</script>

</body>

</html>

1. **Implement the mouseover and mouseout events in simple hyperlink.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Mouse Events</title>

<style>

#mouse-box {

height: 300px;

border: 1px solid black;

background-color: rgb(248, 248, 96);

display: flex;

justify-content: space-around;

align-items: center;

font-size: 40px;

}

</style>

</head>

<body>

<div id="mouse-box">

<a href="#" onmouseover="onMouseOver(this)" onmouseout="onMouseOut(this)"> Hover over me </a>

</div>

</body>

</html>

<script>

function onMouseOver(link) {

link.style.color = "red";

link.textContent = "Mouse over me!";

}

function onMouseOut(link) {

link.style.color = "green";

}

</script>