Module 7

Adv. Javascript

Q1. <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Advanced JavaScript Alert</title>

</head>

<body>

<button id="alertButton">Click Me</button>

<script>

// Advanced JavaScript: Adding Event Listener

document.getElementById("alertButton").addEventListener("click", () => {

const now = new Date().toLocaleTimeString();

alert(`Hello! The current time is: ${now}`);

});

</script>

</body>

</html>

Q2. **5 > 4**

**Result**: true

**"apple" > "pineapple"**

* **Result**: false

**"2" > "12"**

* **Result**: true

**undefined == null**

* **Result**: true

**undefined === null**

* **Result**: false

**null == "\n0\n"**

* **Result**: false

**null === +"\n0\n"**

* **Result**: false

Q3. Will alert be shown? if ("0") { alert( 'Hello'); }

Yes, the alert will be shown, and it will display "Hello."

Q4. function checkAge(age) {

if (age > 18) {

return true; // If age is greater than 18, return true.

} else {

return confirm('Did parents allow you?');

}

}

Q5. const user = {};

user.name = "Vraj";

user.surname = "Patel";

user.name = "Het";

delete user.name;

Q6. <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Password Hide/Show</title>

</head>

<body>

<div class="container">

<label for="password">Password:</label>

<input type="password" id="password" placeholder="Enter your password" />

<label>

<input type="checkbox" id="togglePassword" /> Show Password

</label>

</div>

<script>

const passwordInput = document.getElementById('password');

const togglePassword = document.getElementById('togglePassword');

togglePassword.addEventListener('change', function () {

if (this.checked) {

passwordInput.type = 'text';

}

else {

passwordInput.type = 'password';

}

});

</script>

</body>

</html>

Q7. <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Button Alert on Click</title>

<style>

.btn {

display: inline-block;

margin: 10px;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

}

</style>

</head>

<body>

<div>

<button class="btn">Button 1</button>

<button class="btn">Button 2</button>

<button class="btn">Button 3</button>

<button class="btn">Button 4</button>

</div>

<script>

const buttons = document.querySelectorAll('.btn');

buttons.forEach((button, index) => {

button.addEventListener('click', () => {

alert(`Button ${index + 1} clicked!`);

});

});

</script>

</body>

</html>

Q8. <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Responsive Header</title>

<style>

body {

margin: 0;

font-family: Arial, sans-serif;

}

.header {

display: flex;

justify-content: space-between;

align-items: center;

padding: 10px 20px;

background-color: #333;

color: white;

}

.header h1 {

margin: 0;

}

.nav-links {

display: flex;

list-style: none;

gap: 20px;

}

.nav-links a {

text-decoration: none;

color: white;

}

.hamburger {

display: none;

flex-direction: column;

cursor: pointer;

}

.hamburger div {

width: 25px;

height: 3px;

background-color: white;

margin: 4px 0;

}

@media (max-width: 768px) {

.nav-links {

display: none;

flex-direction: column;

gap: 10px;

background-color: #444;

position: absolute;

top: 50px;

right: 20px;

padding: 10px;

border-radius: 5px;

}

.nav-links.active {

display: flex;

}

.hamburger {

display: flex;

}

}

</style>

</head>

<body>

<header class="header">

<h1>Logo</h1>

<ul class="nav-links">

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Services</a></li>

<li><a href="#">Contact</a></li>

</ul>

<div class="hamburger" id="hamburger">

<div></div>

<div></div>

<div></div>

</div>

</header>

<script>

// Select the hamburger icon and navigation links

const hamburger = document.getElementById('hamburger');

const navLinks = document.querySelector('.nav-links');

// Toggle navigation visibility when the hamburger icon is clicked

hamburger.addEventListener('click', () => {

navLinks.classList.toggle('active');

});

</script>

</body>

</html>

Q9.

HTML file :-

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Form Validation</title>

<style>

.error { color: red; font-size: 0.9em; }

.success { color: green; font-size: 0.9em; }

input { margin-bottom: 10px; display: block; }

</style>

</head>

<body>

<h2>Registration Form</h2>

<form id="registrationForm">

<label for="name">Name:</label>

<input type="text" id="name" name="name" placeholder="Enter your name">

<span class="error" id="nameError"></span>

<label for="email">Email:</label>

<input type="email" id="email" name="email" placeholder="Enter your email">

<span class="error" id="emailError"></span>

<label for="password">Password:</label>

<input type="password" id="password" name="password" placeholder="Enter your password">

<span class="error" id="passwordError"></span>

<label for="confirmPassword">Confirm Password:</label>

<input type="password" id="confirmPassword" name="confirmPassword" placeholder="Re-enter your password">

<span class="error" id="confirmPasswordError"></span>

<button type="submit">Register</button>

<span class="success" id="successMessage"></span>

</form>

<script src="validation.js"></script>

</body>

</html>

JAVASCRIPT FILE :-

document.getElementById('registrationForm').addEventListener('submit', function (e) {

e.preventDefault();

document.querySelectorAll('.error').forEach(el => el.textContent = '');

document.getElementById('successMessage').textContent = '';

const name = document.getElementById('name').value.trim();

const email = document.getElementById('email').value.trim();

const password = document.getElementById('password').value;

const confirmPassword = document.getElementById('confirmPassword').value;

let isValid = true;

if (name === '') {

document.getElementById('nameError').textContent = 'Name is required.';

isValid = false;

} else if (name.length < 3) {

document.getElementById('nameError').textContent = 'Name must be at least 3 characters.';

isValid = false;

}

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (email === '') {

document.getElementById('emailError').textContent = 'Email is required.';

isValid = false;

} else if (!emailRegex.test(email)) {

document.getElementById('emailError').textContent = 'Enter a valid email.';

isValid = false;

}

if (password === '') {

document.getElementById('passwordError').textContent = 'Password is required.';

isValid = false;

} else if (password.length < 6) {

document.getElementById('passwordError').textContent = 'Password must be at least 6 characters.';

isValid = false;

}

// Confirm password validation

if (confirmPassword !== password) {

document.getElementById('confirmPasswordError').textContent = 'Passwords do not match.';

isValid = false;

}

// If all validations pass

if (isValid) {

document.getElementById('successMessage').textContent = 'Registration successful!';

document.getElementById('registrationForm').reset(); // Reset form fields

}

});

Q10.

JAVASCRIPT Code:

document.getElementById('registrationForm').addEventListener('submit', function (e) {

e.preventDefault(); // Prevent the default form submission (refreshing the page)

document.querySelectorAll('.error').forEach(el => el.textContent = '');

document.getElementById('successMessage').textContent = '';

const name = document.getElementById('name').value.trim();

const email = document.getElementById('email').value.trim();

const password = document.getElementById('password').value;

const confirmPassword = document.getElementById('confirmPassword').value;

let isValid = true;

if (name === '') {

document.getElementById('nameError').textContent = 'Name is required.';

isValid = false;

} else if (name.length < 3) {

document.getElementById('nameError').textContent = 'Name must be at least 3 characters.';

isValid = false;

}

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (email === '') {

document.getElementById('emailError').textContent = 'Email is required.';

isValid = false;

} else if (!emailRegex.test(email)) {

document.getElementById('emailError').textContent = 'Enter a valid email.';

isValid = false;

}

if (password === '') {

document.getElementById('passwordError').textContent = 'Password is required.';

isValid = false;

} else if (password.length < 6) {

document.getElementById('passwordError').textContent = 'Password must be at least 6 characters.';

isValid = false;

}

if (confirmPassword !== password) {

document.getElementById('confirmPasswordError').textContent = 'Passwords do not match.';

isValid = false;

}

if (isValid) {

document.getElementById('successMessage').textContent = 'Registration successful!';

document.getElementById('registrationForm').reset(); // Reset form fields

}

});

Q11. <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Product List</title>

<style>

.product-item {

border: 1px solid #ccc;

padding: 10px;

margin: 10px 0;

border-radius: 5px;

max-width: 400px;

}

.product-item img {

display: block;

margin-bottom: 10px;

}

.error { color: red; }

</style>

</head>

<body>

<h2>Product List</h2>

<div id="productList">Loading products...</div>

<script src="fetch-products.js"></script>

</body>

</html>

document.addEventListener('DOMContentLoaded', function () {

// Fetch products from the API

fetch('https://fakestoreapi.com/products')

.then(response => {

if (!response.ok) {

throw new Error('Network response was not ok ' + response.statusText);

}

return response.json();

})

.then(products => {

displayProducts(products); // Call the function to display products

})

.catch(error => {

console.error('There was a problem with the fetch operation:', error);

document.getElementById('productList').innerHTML = `<p class="error">Failed to load products. Please try again later.</p>`;

});

});

// Function to display products in the DOM

function displayProducts(products) {

const productList = document.getElementById('productList');

productList.innerHTML = ''; // Clear previous content

products.forEach(product => {

const productItem = document.createElement('div');

productItem.classList.add('product-item');

productItem.innerHTML = `

<h3>${product.title}</h3>

<img src="${product.image}" alt="${product.title}" width="100">

<p>Price: $${product.price}</p>

<p>${product.description.substring(0, 100)}...</p>

`;

productList.appendChild(productItem);

});

}

Q12.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Responsive Header</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<div class="logo">

<h1>My Logo</h1>

</div>

<nav class="navbar">

<ul id="nav-links">

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Services</a></li>

<li><a href="#">Contact</a></li>

</ul>

<div class="hamburger" id="hamburger">

<span class="bar"></span>

<span class="bar"></span>

<span class="bar"></span>

</div>

</nav>

</header>

<script src="script.js"></script>

</body>

</html>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: Arial, sans-serif;

line-height: 1.6;

}

header {

display: flex;

justify-content: space-between;

align-items: center;

padding: 1rem 2rem;

background-color: #333;

color: #fff;

}

.logo h1 {

font-size: 1.8rem;

}

.navbar ul {

display: flex;

list-style-type: none;

}

.navbar li {

margin: 0 1rem;

}

.navbar a {

color: white;

text-decoration: none;

font-size: 1rem;

}

.hamburger {

display: none;

flex-direction: column;

cursor: pointer;

}

.hamburger .bar {

width: 25px;

height: 4px;

background-color: white;

margin: 4px 0;

}

/\* Media Query \*/

@media screen and (max-width: 768px) {

.navbar ul {

display: none;

width: 100%;

text-align: center;

position: absolute;

top: 70px;

left: 0;

background-color: #333;

padding: 1rem 0;

}

.navbar li {

margin: 1rem 0;

}

.hamburger {

display: flex;

}

.navbar.active ul {

display: block;

}

}

const hamburger = document.getElementById('hamburger');

const navLinks = document.getElementById('nav-links');

hamburger.addEventListener('click', () => {

navLinks.classList.toggle('active');

});

Q13.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Modal Box</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<!-- Button to Open Modal -->

<button id="openModalBtn">Open Modal</button>

<!-- Modal -->

<div id="myModal" class="modal">

<div class="modal-content">

<span class="close" id="closeModalBtn">&times;</span>

<h2>Modal Heading</h2>

<p>This is a simple modal box with three buttons.</p>

<button id="button1">Button 1</button>

<button id="button2">Button 2</button>

<button id="button3">Button 3</button>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

/\* General styles for body \*/

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f4f4f4;

margin: 0;

}

/\* Button to trigger modal \*/

#openModalBtn {

padding: 10px 20px;

font-size: 16px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

#openModalBtn:hover {

background-color: #45a049;

}

/\* The Modal (background) \*/

.modal {

display: none; /\* Hidden by default \*/

position: fixed;

z-index: 1; /\* Sit on top \*/

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgb(0,0,0); /\* Black w/ opacity \*/

background-color: rgba(0, 0, 0, 0.4); /\* Black w/ opacity \*/

}

/\* Modal Content \*/

.modal-content {

background-color: #fff;

margin: 15% auto;

padding: 20px;

border-radius: 10px;

width: 40%;

text-align: center;

}

/\* The Close Button \*/

.close {

color: #aaa;

font-size: 28px;

font-weight: bold;

position: absolute;

top: 10px;

right: 20px;

cursor: pointer;

}

.close:hover,

.close:focus {

color: black;

text-decoration: none;

cursor: pointer;

}

/\* Style for the buttons inside the modal \*/

button {

padding: 10px 20px;

margin: 10px;

font-size: 16px;

cursor: pointer;

border: none;

border-radius: 5px;

}

button:hover {

background-color: #ddd;

}

/\* Style for individual buttons \*/

#button1 {

background-color: #4CAF50;

color: white;

}

#button2 {

background-color: #2196F3;

color: white;

}

#button3 {

background-color: #f44336;

color: white;

}

// Get elements

const modal = document.getElementById('myModal');

const openModalBtn = document.getElementById('openModalBtn');

const closeModalBtn = document.getElementById('closeModalBtn');

// Get buttons inside modal

const button1 = document.getElementById('button1');

const button2 = document.getElementById('button2');

const button3 = document.getElementById('button3');

// Open the modal when the button is clicked

openModalBtn.onclick = function() {

modal.style.display = "block";

}

// Close the modal when the close button (X) is clicked

closeModalBtn.onclick = function() {

modal.style.display = "none";

}

// Close the modal if user clicks anywhere outside of the modal content

window.onclick = function(event) {

if (event.target == modal) {

modal.style.display = "none";

}

}

// Action for Button 1

button1.onclick = function() {

alert('Button 1 clicked!');

}

// Action for Button 2

button2.onclick = function() {

alert('Button 2 clicked!');

}

// Action for Button 3

button3.onclick = function() {

alert('Button 3 clicked!');

}

Q14.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Swiper Slider Example</title>

<!-- Swiper CSS (from CDN) -->

<link rel="stylesheet" href="https://unpkg.com/swiper/swiper-bundle.min.css">

<style>

/\* Optional: Additional custom styling for the slider \*/

.swiper-container {

width: 100%;

height: 400px;

margin: 20px auto;

}

.swiper-slide {

display: flex;

justify-content: center;

align-items: center;

background-color: #ccc;

font-size: 2rem;

color: white;

}

</style>

</head>

<body>

<!-- Swiper Slider -->

<div class="swiper-container">

<div class="swiper-wrapper">

<div class="swiper-slide">Slide 1</div>

<div class="swiper-slide">Slide 2</div>

<div class="swiper-slide">Slide 3</div>

<div class="swiper-slide">Slide 4</div>

</div>

<!-- Pagination, Navigation Buttons -->

<div class="swiper-pagination"></div>

<div class="swiper-button-next"></div>

<div class="swiper-button-prev"></div>

<div class="swiper-scrollbar"></div>

</div>

<!-- Swiper JS (from CDN) -->

<script src="https://unpkg.com/swiper/swiper-bundle.min.js"></script>

<script>

// Initialize Swiper

const swiper = new Swiper('.swiper-container', {

loop: true, // Loop the slides

slidesPerView: 1, // Number of slides visible at once

spaceBetween: 10, // Space between slides

navigation: {

nextEl: '.swiper-button-next',

prevEl: '.swiper-button-prev',

},

pagination: {

el: '.swiper-pagination',

clickable: true,

},

scrollbar: {

el: '.swiper-scrollbar',

draggable: true,

},

});

</script>

</body>

</html>

Q15.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Fetch Products</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Product List</h1>

<div id="product-container" class="product-container">

<!-- Products will be displayed here -->

</div>

<script src="script.js"></script>

</body>

</html>

body {

font-family: Arial, sans-serif;

background-color: #f9f9f9;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

h1 {

text-align: center;

margin-bottom: 20px;

}

.product-container {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));

gap: 20px;

padding: 20px;

}

.product-card {

background-color: #fff;

border-radius: 10px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

padding: 20px;

text-align: center;

}

.product-card img {

width: 100%;

height: auto;

border-radius: 8px;

}

.product-card h3 {

font-size: 1.2rem;

margin: 15px 0 10px;

}

.product-card p {

font-size: 1rem;

color: #555;

}

.product-card .price {

font-size: 1.1rem;

font-weight: bold;

color: #4CAF50;

}

const productContainer = document.getElementById('product-container');

fetch('https://fakestoreapi.com/products')

.then(response => response.json()) // Parse the JSON response

.then(data => {

data.forEach(product => {

const productCard = document.createElement('div');

productCard.classList.add('product-card');

productCard.innerHTML = `

<img src="${product.image}" alt="${product.title}">

<h3>${product.title}</h3>

<p>${product.description.substring(0, 100)}...</p>

<p class="price">$${product.price}</p>

`;

productContainer.appendChild(productCard);

});

})

.catch(error => {

console.error('Error fetching the products:', error);

productContainer.innerHTML = '<p>Failed to load products. Please try again later.</p>';

});