# Lab 1 [Getting started with HTML, CSS & JavaScript]

|  |  |
| --- | --- |
| Lab Title: | Getting started with HTML, CSS & JavaScript |
| **Expected duration** (hours): | **1 hour 45 mins** |

|  |
| --- |
| Objectives |
| Understanding basic HTML, CSS & JavaScript syntax |
| Validate data in a webpage using JavaScript |
| Learn how to make a webpage interactive |

|  |
| --- |
| Requirements |
| A laptop or a desktop with Windows or Mac as an operating system |
| Visual studio code or something similar as a text editor |
| A modern web browser like chrome |

## Lab Instructions:

# Understanding HTML, CSS, JavaScript and create basic web pages

This lab will help you to get familiar with basic HTML, CSS and JavaScript syntax, you will add a product list and created product page and then transform the create product page interactive. None of these pages will use anything but HTML, CSS and JavaScript. The purpose of this lab is to getting students familiar with a scripting language like JavaScript and show the power of it.

### **Part 1 - Style a list product page using CSS**

1. Open your text editor or IDE (such as Visual Studio Code, Sublime Text, or any code editor of your choice).
2. Create a new file and name it product\_list.html.
3. Copy and paste the following HTML code into your product\_list.html file:

<!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>

</head>

<body>

<div>

<h1>Available Products</h1>

<div>

<h2>Smartphone XYZ</h2>

<p>A high-end smartphone with a 6.5-inch display, 128GB storage, and a 48MP camera.</p>

<p>Price: $799</p>

</div>

<div>

<h2>Laptop Pro 15</h2>

<p>A powerful laptop with a 15-inch screen, 16GB RAM, and 512GB SSD for professionals.</p>

<p>Price: $1,299</p>

</div>

<div>

<h2>Wireless Earbuds</h2>

<p>Compact and comfortable wireless earbuds with noise cancellation and 20-hour battery life.</p>

<p>Price: $199</p>

</div>

<div>

<h2>4K Ultra HD TV</h2>

<p>A stunning 65-inch 4K Ultra HD TV with HDR and smart features for streaming.</p>

<p>Price: $999</p>

</div>

<div>

<h2>Smartwatch Series 5</h2>

<p>A versatile smartwatch with fitness tracking, GPS, and heart rate monitoring.</p>

<p>Price: $299</p>

</div>

</div>

</body>

</html>

1. Save the product\_list.html file.
2. Create a new directory named css at the same level as html
3. Create a new file inside the css directory and name it styles.css.
4. Copy and paste the following CSS code into your styles.css file:

/\* styles.css \*/

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

.container {

width: 80%;

margin: 0 auto;

padding: 20px;

background-color: #fff;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1 {

text-align: center;

color: #333;

}

.product {

background: #eee;

margin: 10px 0;

padding: 15px;

border-radius: 5px;

}

.product h2 {

margin: 0;

color: #333;

}

.product p {

margin: 5px 0;

color: #666;

}

.product .instructor {

font-weight: bold;

}

1. Save the styles.css file.
2. Update the html page with relevant classes as below <!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>

<link rel="stylesheet" href="../css/styles.css">

</head>

<body>

<div class="container">

<h1>Available Products</h1>

<div class="product">

<h2>Smartphone X100</h2>

<p>A high-end smartphone with a sleek design, powerful processor, and advanced camera features.</p>

<p class="price">$999</p>

</div>

<div class="product">

<h2>4K Ultra HD TV</h2>

<p>An ultra-high-definition television with stunning picture quality and smart features for streaming.</p>

<p class="price">$799</p>

</div>

<div class="product">

<h2>Wireless Noise-Cancelling Headphones</h2>

<p>Premium headphones with noise-cancelling technology and exceptional sound quality.</p>

<p class="price">$299</p>

</div>

<div class="product">

<h2>Laptop Pro 15</h2>

<p>A powerful laptop with a high-resolution display, fast processor, and ample storage.</p>

<p class="price">$1,299</p>

</div>

<div class="product">

<h2>Smartwatch Series 5</h2>

<p>A versatile smartwatch with fitness tracking, notifications, and customizable watch faces.</p>

<p class="price">$399</p>

</div>

</div>

</body>

</html>

1. Open the product\_list.html file in your web browser by double-clicking the file or dragging it into an open browser window.
2. Observe the styled product list with a container, headings, paragraphs, and instructor information.
3. Let's break down the CSS rules and see how they apply to the HTML elements:
   1. Styling the body:

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

* font-family: Arial, sans-serif; sets the font for the entire page to Arial with a fallback to sans-serif.
* background-color: #f4f4f4; sets a light grey background color for the entire page.
* margin: 0; removes the default margin around the page.
* padding: 0; removes the default padding around the page.
  1. Styling the container:

.container {

width: 80%;

margin: 0 auto;

padding: 20px;

background-color: #fff;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

* width: 80%; sets the width of the container to 80% of the page width.
* margin: 0 auto; centers the container horizontally.
* padding: 20px; adds padding inside the container.
* background-color: #fff; sets a white background color for the container.
* box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); adds a subtle shadow around the container to give it a raised appearance.
  1. Styling the main heading:

h1 {

text-align: center;

color: #333;

}

* text-align: center; centers the heading text.
* color: #333; sets the text color to a dark grey.
  1. Styling each product block:

.product {

background: #eee;

margin: 10px 0;

padding: 15px;

border-radius: 5px;

}

* background: #eee; sets a light grey background color for each product block.
* margin: 10px 0; adds margin above and below each product block.
* padding: 15px; adds padding inside each product block.
* border-radius: 5px; rounds the corners of each product block.
  1. Styling the product titles:

.product h2 {

margin: 0;

color: #333;

}

* margin: 0; removes the default margin around the product titles.
* color: #333; sets the text color to dark grey.
  1. Styling the product descriptions:

.product p {

margin: 5px 0;

color: #666;

}

* margin: 5px 0; adds margin above and below the product descriptions.
* color: #666; sets the text color to a medium grey.
  1. Highlighting the instructor's name:

.product .instructor {

font-weight: bold;

}

* font-weight: bold; makes the instructor's name bold to stand out.

### **Part 2 - Style a list product page using CSS**

1. Open your text editor or IDE (such as Visual Studio Code, Sublime Text, or any code editor of your choice).
2. Create a new file and name it create\_product.html.
3. Copy and paste the following HTML code into your create\_product.html file: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Create a Product</title>

</head>

<body>

<div>

<h1>Create a Product</h1>

<form>

<label for="productName">Product Name</label>

<input type="text" id="productName" name="productName" required>

<br/>

<br/>

<label for="productDescription">Product Description</label>

<textarea id="productDescription" name="productDescription" rows="5" required></textarea>

<br/>

<br/>

<label for="price">Price</label>

<input type="text" id="price" name="price" required>

<br/>

<br/>

<button type="submit">Create Product</button>

</form>

</div>

</body>

</html>

1. Save the create\_product.html file.
2. Create a new file inside the css directory and name it form-styles.css.
3. Copy and paste the following CSS code into your form-styles.css file:

/\* form-styles.css \*/

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

.container {

width: 50%;

margin: 50px auto;

padding: 20px;

background-color: #fff;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

border-radius: 8px;

}

h1 {

text-align: center;

color: #333;

}

form {

display: flex;

flex-direction: column;

}

label {

margin: 10px 0 5px;

font-weight: bold;

}

input, textarea {

padding: 10px;

margin: 5px 0 20px;

border: 1px solid #ccc;

border-radius: 4px;

width: 100%;

}

button {

padding: 10px;

color: #fff;

background-color: #333;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #555;

}

1. Save the form-styles.css file.
2. Update the html page with relevant classes as below <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Create a Product</title>

<link rel="stylesheet" href="../css/form-styles.css">

</head>

<body>

<div class="container">

<h1>Create a Product</h1>

<form>

<label for="productName">Product Name</label>

<input type="text" id="productName" name="productName" required>

<label for="productDescription">Product Description</label>

<textarea id="productDescription" name="productDescription" rows="5" required></textarea>

<label for="productPrice">Product Price</label>

<input type="number" id="productPrice" name="productPrice" required>

<button type="submit">Create Product</button>

</form>

</div>

</body>

</html>

1. Open the create\_product.html file in your web browser by double-clicking the file or dragging it into an open browser window.
2. Observe the styled create product page.

### **Part 3 - Style a home page using CSS**

1. Open your text editor or IDE (such as Visual Studio Code, Sublime Text, or any code editor of your choice).
2. Create a new file and name it home\_page.html.
3. Copy and paste the following HTML code into your home\_page.html file:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Home Page</title>

<link rel="stylesheet" href="../css/home-styles.css">

</head>

<body>

<div class="container">

<h1>Welcome to the Product Management System</h1>

<a href="product\_list.html" class="button">View Products</a>

<a href="create\_product.html" class="button">Create a Product</a>

</div>

</body>

</html>

1. Save the home\_page.html file.
2. Create a new file inside the css directory and name it form-styles.css.
3. Copy and paste the following CSS code into your form-styles.css file:

/\* home-styles.css \*/

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

.container {

width: 80%;

margin: 50px auto;

padding: 20px;

background-color: #fff;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

text-align: center;

}

h1 {

color: #333;

}

.button {

display: inline-block;

margin: 20px;

padding: 15px 25px;

color: #fff;

background-color: #333;

text-decoration: none;

border-radius: 5px;

transition: background-color 0.3s;

}

.button:hover {

background-color: #555;

}

1. Save the home-styles.css file.
2. Update the html page with relevant classes as below

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Home Page</title>

<link rel="stylesheet" href="../css/home-styles.css">

</head>

<body>

<div class="container">

<h1>Welcome to the Product Management System</h1>

<a href="product\_list.html" class="button">View Products</a>

<a href="create\_product.html" class="button">Create a Product</a>

</div>

</body>

</html>

1. Open the home\_page.html file in your web browser by double-clicking the file or dragging it into an open browser window.
2. Observe the styled home page with container and button class.

### Part 4 - Update create product page using JavaScript

1. Set Up Your Environment
   1. Open your preferred code editor (such as VS Code, Sublime Text, or Notepad++).
   2. Open a new file named create\_product.html.
2. Basic HTML Structure
   1. Make sure the below HTML structure with CSS exists in the create\_product.html page.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Create a Product</title>

<link rel="stylesheet" href="../css/form-styles.css">

</head>

<body>

<div class="container">

<h1>Create a Product</h1>

<form>

<label for="productName">Product Name</label>

<input type="text" id="productName" name="productName" required>

<label for="productDescription">Product Description</label>

<textarea id="productDescription" name="productDescription" rows="5" required></textarea>

<label for="productPrice">Product Price</label>

<input type="number" id="productPrice" name="productPrice" required>

<button type="submit">Create Product</button>

</form>

</div>

</body>

</html>

1. Add JavaScript for Form Validation
   1. Below the closing </form> tag, add a <script> tag to include your JavaScript.

<script>

document.getElementById('productForm').addEventListener('submit', function(event) {

event.preventDefault();

// Get form values

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

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

const productPrice = parseFloat(document.getElementById('productPrice').value);

// Validation flags

let isValid = true;

// Validate Product Name

if (productName.length === 0) {

isValid = false;

alert('Product Name is required.');

} else if (productName.length < 3 || productName.length > 50) {

isValid = false;

alert('Product Name must be between 3 and 50 characters.');

}

// Validate Product Description

if (productDescription.length === 0) {

isValid = false;

alert('Product Description is required.');

} else if (productDescription.length < 10 || productDescription.length > 200) {

isValid = false;

alert('Product Description must be between 10 and 200 characters.');

}

// Validate Product Price

if (isNaN(productPrice) || productPrice <= 0) {

isValid = false;

alert('Product Price must be a positive number.');

}

// If all validations pass, submit the form

if (isValid) {

alert('Product created successfully!');

// You can add code here to submit the form data to the server

}

});

</script>

1. Understand the JavaScript Code
   1. The JavaScript code does the following:
      1. Listens for the form submission event.
      2. Prevents the default form submission.
      3. Retrieves and trims the form values.
      4. Validates the form fields:
         1. Checks if the product name is between 3 and 50 characters.
         2. Ensures the product description is between 10 and 200 characters.
         3. Validates the product price to ensure it is a positive number.
      5. If all validations pass, it alerts the user that the product has been created successfully.
2. Enhance User Feedback
   1. Instead of using alert, you can enhance the user experience by displaying validation messages directly in the form. Update the HTML structure to include message containers:

<div class="container">

<h1>Create a Product</h1>

<form id="productForm">

<label for="productName">Product Name</label>

<input type="text" id="productName" name="productName" required>

<div id="productNameError" class="error"></div>

<label for="productDescription">Product Description</label>

<textarea id="productDescription" name="productDescription" rows="5" required></textarea>

<div id="productDescriptionError" class="error"></div>

<label for="productPrice">Product Price</label>

<input type="number" id="productPrice" name="productPrice" required>

<div id="productPriceError" class="error"></div>

<button type="submit">Create Product</button>

</form>

</div>

1. Update JavaScript for Enhanced Feedback
   1. Modify the JavaScript to display error messages in the form:

document.getElementById('productForm').addEventListener('submit', function(event) {

event.preventDefault();

// Get form values

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

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

const productPrice = parseFloat(document.getElementById('productPrice').value);

// Get error message containers

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

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

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

// Clear previous error messages

productNameError.textContent = '';

productDescriptionError.textContent = '';

productPriceError.textContent = '';

// Validation flags

let isValid = true;

// Validate Product Name

if (productName.length === 0) {

isValid = false;

productNameError.textContent = 'Product Name is required.';

} else if (productName.length < 3 || productName.length > 50) {

isValid = false;

productNameError.textContent = 'Product Name must be between 3 and 50 characters.';

}

// Validate Product Description

if (productDescription.length === 0) {

isValid = false;

productDescriptionError.textContent = 'Product Description is required.';

} else if (productDescription.length < 10 || productDescription.length > 200) {

isValid = false;

productDescriptionError.textContent = 'Product Description must be between 10 and 200 characters.';

}

// Validate Product Price

if (isNaN(productPrice) || productPrice <= 0) {

isValid = false;

productPriceError.textContent = 'Product Price must be a positive number.';

}

// If all validations pass, submit the form

if (isValid) {

alert('Product created successfully!');

// You can add code here to submit the form data to the server

}

});