



# Apps Script **Code Exercises**

 Transform Your  
Google Apps Script  
Projects with HTML  
Service! 





<b>Apps Script HTML Service Coding Exercises</b>	<b>1</b>
<b>Exercise 1: Create and Display a Basic HTML Page</b>	<b>2</b>
<b>Exercise 2: Embedding CSS in HTML Page</b>	<b>3</b>
<b>Exercise 3: Using JavaScript in HTML Page</b>	<b>4</b>
<b>Exercise 4: Passing Data from Google Apps Script to HTML</b>	<b>5</b>
<b>Exercise 5: Creating a Form and Handling Submissions</b>	<b>5</b>
<b>Exercise 6: Fetching Data from External APIs in HTML Page</b>	<b>7</b>
<b>Exercise 7: Using Templated HTML</b>	<b>8</b>
<b>Exercise 8: Communicating between HTML Page and Google Apps Script</b>	<b>9</b>
<b>Exercise 9: Handling Events in Client-Side JavaScript</b>	<b>10</b>
<b>Exercise 10: Dynamic UI Updates with JavaScript</b>	<b>11</b>


## Apps Script HTML Service Coding Exercises

 Transform Your Google Apps Script Projects with HTML Service! 

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

 Dive into 10 engaging coding exercises to master the HTML Service in Google Apps Script! These exercises will guide you through creating dynamic HTML pages, styling them with CSS, making them interactive with JavaScript, and integrating them seamlessly with Google Sheets.

 From basic HTML pages to advanced dynamic content and API integrations, these exercises are designed to enhance your web development skills within the Google Apps Script environment.

 Unleash the full potential of your Google Apps Script projects with these hands-on exercises!

## Exercise 1: Create and Display a Basic HTML Page

Objective: Learn to create and display a simple HTML page using HtmlService.

Explanation: This exercise introduces the basic operation of creating and serving a simple HTML page.

Code:

```
function showBasicHtmlPage() {  
  var html = "<html><body><h1>Hello World</h1></body></html>";  
  var output = HtmlService.createHtmlOutput(html);
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```
SpreadsheetApp.getUi().showModalDialog(output, "My Page");  
}
```

## Exercise 2: Embedding CSS in HTML Page

Objective: Understand how to style an HTML page using CSS.

Explanation: This exercise focuses on adding CSS to an HTML page to enhance its appearance.

Code:

```
function showStyledHtmlPage() {  
  var html = `  
    <html>  
      <head>  
        <style>  
          body { font-family: Arial; }  
          h1 { color: blue; }  
        </style>  
      </head>  
      <body>  
        <h1>Styled HTML Page</h1>  
      </body>  
    </html>`;  
  var output = HtmlService.createHtmlOutput(html);
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```
SpreadsheetApp.getUi().showModalDialog(output, "Styled Page");  
}
```

## Exercise 3: Using JavaScript in HTML Page

Objective: Learn to make an HTML page interactive with JavaScript.

Explanation: This exercise teaches how to use JavaScript to add interactivity to an HTML page.

Code:

```
function showHtmlPageWithJs() {  
  var html = `  
    <html>  
      <body>  
        <button onclick="alert('Hello!')">Click Me</button>  
      </body>  
    </html>`;  
  var output = HtmlService.createHtmlOutput(html);  
  SpreadsheetApp.getUi().showModalDialog(output, "Interactive Page");  
}
```

## Exercise 4: Passing Data from Google Apps Script to HTML

Objective: Understand how to pass data from a Google Apps Script to an HTML page.

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

Explanation: This exercise demonstrates how to send data from your script to be displayed on an HTML page.

Code:

```
function showHtmlPageWithData() {  
  var data = { name: "Alice", email: "alice@example.com" };  
  var html = HtmlService.createTemplate('<p>Name: <?= name ?></p><p>Email:  
<?= email ?></p>');  
  html.name = data.name;  
  html.email = data.email;  
  var output = html.evaluate();  
  SpreadsheetApp.getUi().showModalDialog(output, "Page With Data");  
}
```

## Exercise 5: Creating a Form and Handling Submissions

Objective: Learn to create a form in an HTML page and handle submissions.

Explanation: This exercise teaches how to create an HTML form and process the data submitted by the user.

Code:

```
function showFormPage() {  
  var html = `
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```

<html>
  <body>
    <form id="myForm">
      <input type="text" name="input">
      <input type="submit" value="Submit">
    </form>
    <script>
      document.getElementById('myForm').addEventListener('submit', function(e)
{
      e.preventDefault();
      google.script.run.processForm(this);
    });
  </script>
</body>
</html>`;

var output = HtmlService.createHtmlOutput(html);
SpreadsheetApp.getUi().showModalDialog(output, "Form Page");
}

function processForm(formObject) {
  Logger.log('Form input: ' + formObject.input);
}

```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

## Exercise 6: Fetching Data from External APIs in HTML Page

Objective: Understand how to fetch data from an external API and display it in an HTML page.

Explanation: This exercise demonstrates how to make a call to an external API from an HTML page and display the data.

Code:

```
function showApiDataPage() {
  var html = `
  <html>
  <body>
    <button onclick="loadData()">Load Data</button>
    <div id="data"></div>
    <script>
      function loadData() {
        fetch('https://api.example.com/data') // Replace with actual API URL
        .then(response => response.json())
        .then(data => {
          document.getElementById('data').innerText = JSON.stringify(data);
        });
      }
    </script>
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```
    </body>
  </html>`;
  var output = HtmlService.createHtmlOutput(html);
  SpreadsheetApp.getUi().showModalDialog(output, "API Data Page");
}
```

## Exercise 7: Using Templated HTML

Objective: Learn to use Google Apps Script templating features to generate dynamic HTML content.

Explanation: This exercise teaches how to create dynamic HTML content based on script data using templating.

Code:

```
function showTemplatedHtmlPage() {
  var data = { title: "My Page", items: ["One", "Two", "Three"] };
  var template = HtmlService.createTemplateFromFile('PageTemplate'); // Assumes
  an HTML file named 'PageTemplate' exists
  template.data = data;
  var output = template.evaluate();
  SpreadsheetApp.getUi().showModalDialog(output, data.title);
}
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>



## Exercise 8: Communicating between HTML Page and Google Apps Script

Objective: Understand the communication between an HTML page and the server-side Google Apps Script.

Explanation: This exercise demonstrates how client-side JavaScript can interact with server-side Google Apps Script functions.

Code:

```
function showCommunicationPage() {
  var html = `
  <html>
  <body>
    <button onclick="sendMessageToServer()">Send Message</button>
  <script>
    function sendMessageToServer() {
      google.script.run.withSuccessHandler(onSuccess).sendMessage('Hello from
Client!');
    }

    function onSuccess(response) {
      alert(response);
    }
  </script>
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```

    </body>
</html>`;
var output = HtmlService.createHtmlOutput(html);
SpreadsheetApp.getUi().showModalDialog(output, "Communication Page");
}

function sendMessage(message) {
    Logger.log(message);
    return "Message received: " + message;
}

```

## Exercise 9: Handling Events in Client-Side JavaScript

Objective: Learn to handle various events in the HTML page using JavaScript.

Explanation: This exercise focuses on handling different types of user interactions on an HTML page.

Code:

```

function showEventHandlingPage() {
    var html = `
    <html>
    <body>
        <input type="text" id="myInput">
        <script>

```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```

    document.getElementById('myInput').addEventListener('change', function()
{
    alert('Text changed to: ' + this.value);
    });
</script>
</body>
</html>`;
var output = HtmlService.createHtmlOutput(html);
SpreadsheetApp.getUi().showModalDialog(output, "Event Handling Page");
}

```

## Exercise 10: Dynamic UI Updates with JavaScript

Objective: Understand how to dynamically update the HTML page content using JavaScript.

Explanation: This exercise shows how to modify the content of an HTML page in real-time based on user actions.

Code:

```

function showDynamicUpdatePage() {
    var html = `
    <html>
    <body>
    <div id="content">Original Content</div>

```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```
<button onclick="updateContent()">Update</button>
<script>
  function updateContent() {
    document.getElementById('content').innerText = 'Updated Content';
  }
</script>
</body>
</html>`;
var output = HtmlService.createHtmlOutput(html);
SpreadsheetApp.getUi().showModalDialog(output, "Dynamic Update Page");
}
```