

JavaScript Code Tip Calculator Example

Tip Calculator

Bill Amount:

Service Quality:

Calculate Tip

Tip: \$40.00 | Total: \$440.00

Project: Tip Calculator

Step 1: HTML Structure

Create an HTML file named index.html and set up the basic structure.

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

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Tip Calculator</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Tip Calculator</h1>
    <label for="billAmount">Bill Amount:</label>
    <input type="number" id="billAmount"
step="0.01">
    <label for="serviceQuality">Service
Quality:</label>
    <select id="serviceQuality">
      <option value="0.2">Excellent
(20%)</option>
      <option value="0.15">Good (15%)</option>
      <option value="0.1">Fair (10%)</option>
```

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

```
        <option value="0.05">Poor (5%)</option>
    </select>
    <button id="calculateButton">Calculate
Tip</button>
    <p id="totalTip"></p>
</div>
<script src="script.js"></script>
</body>
</html>
```

Step 2: CSS Styling

Create a CSS file named styles.css for basic styling.

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
```

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

```
}
```

```
.container {  
    background-color: #fff;  
    padding: 20px;  
    border-radius: 5px;  
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
    width: 300px;  
    text-align: center;  
}
```

```
h1 {  
    font-size: 24px;  
    margin-bottom: 10px;  
}
```

```
label {  
    display: block;  
    margin-top: 10px;  
    font-weight: bold;  
}
```

```
input, select {
```

```
width: 100%;
padding: 8px;
margin-top: 5px;
border: 1px solid #ccc;
border-radius: 3px;
}

button {
    display: block;
    margin: 10px auto;
    padding: 8px 16px;
    background-color: #007bff;
    color: #fff;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

#totalTip {
    font-size: 18px;
    font-weight: bold;
    margin-top: 10px;
}
```

Step 3: JavaScript Logic

Create a JavaScript file named script.js for the application logic.

```
const calculateButton =
document.getElementById("calculateButton");
const billAmountInput =
document.getElementById("billAmount");
const serviceQualityInput =
document.getElementById("serviceQuality");
const totalTipElement =
document.getElementById("totalTip");

calculateButton.addEventListener("click",
calculateTip);

function calculateTip() {
    const billAmount =
parseFloat(billAmountInput.value);
    const serviceQuality =
parseFloat(serviceQualityInput.value);
```

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

```
    if (!isNaN(billAmount) && !isNaN(serviceQuality)) {
        const tipAmount = billAmount * serviceQuality;
        const totalAmount = billAmount + tipAmount;

        totalTipElement.textContent = `Tip:
    ${tipAmount.toFixed(2)} | Total:
    ${totalAmount.toFixed(2)}`;
    } else {
        totalTipElement.textContent = "Please enter
valid values.";
    }
}
```

Step 4: Testing

Open the index.html file in a web browser. You should see a Tip Calculator with input fields for the bill amount and service quality, as well as a button to calculate the tip.

Congratulations! You've successfully created a simple Tip Calculator using HTML, CSS, and JavaScript. This project showcases how you can interact with user input and perform calculations on a webpage.

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

Here's the detailed breakdown of the JavaScript code:

We start by getting references to the HTML elements we'll be interacting with: `calculateButton`, `billAmountInput`, `serviceQualityInput`, and `totalTipElement`.

We attach an event listener to the "Calculate Tip" button (`calculateButton`). This listener is set to call the `calculateTip` function when the button is clicked.

The `calculateTip` function is defined. This function:

- Retrieves the values entered by the user for the bill amount and service quality.
- Checks if the entered values are valid numbers using `isNaN()`.
- If the values are valid, it calculates the tip amount and the total amount (bill amount + tip amount).
- Formats and displays the tip amount and total amount in the `totalTipElement`.

By following these steps, the code creates a simple Tip Calculator that calculates the tip and total amount based on the user's input. This project demonstrates how JavaScript can be used to perform calculations and dynamically update content on a webpage.

Step 1: Get References to HTML Elements

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


```
const calculateButton = document.getElementById("calculateButton");
const billAmountInput = document.getElementById("billAmount");
const serviceQualityInput = document.getElementById("serviceQuality");
const totalTipElement = document.getElementById("totalTip");
```

In this step, we're using the `document.getElementById()` method to retrieve references to various HTML elements we'll be interacting with:

- `calculateButton`: The "Calculate Tip" button.
- `billAmountInput`: The input field where the user enters the bill amount.
- `serviceQualityInput`: The dropdown select field where the user selects the service quality.
- `totalTipElement`: The paragraph element where we'll display the calculated tip and total amount.

Step 2: Attach Event Listener

```
calculateButton.addEventListener("click",
calculateTip);
```

Here, we're attaching an event listener to the "Calculate Tip" button. The `calculateTip` function will be called when the button is clicked.

Step 3: Define the calculateTip Function

```
function calculateTip() {
    const billAmount =
parseFloat(billAmountInput.value);
    const serviceQuality =
parseFloat(serviceQualityInput.value);

    if (!isNaN(billAmount) && !isNaN(serviceQuality)) {
        const tipAmount = billAmount * serviceQuality;
        const totalAmount = billAmount + tipAmount;

        totalTipElement.textContent = `Tip:
${tipAmount.toFixed(2)} | Total:
${totalAmount.toFixed(2)}`;
    } else {
        totalTipElement.textContent = "Please enter
valid values.";
    }
}
```

Here's a detailed breakdown of the calculateTip function:

- We start by using parseFloat() to convert the values entered in the billAmountInput and serviceQualityInput fields from strings to floating-point numbers. This allows us to perform calculations with them.
- We use isNaN() to check if the entered values are valid numbers. If both the bill amount and service quality are valid numbers, we proceed with the calculations. If not, we display an error message in the totalTipElement.
- Inside the calculation block:
 - We calculate the tip amount by multiplying the bill amount by the service quality. This gives us the tip amount in dollars.
 - We calculate the total amount by adding the bill amount and the tip amount. This gives us the total amount including the tip.
- We use the toFixed(2) method to format the calculated tip and total amounts to two decimal places, ensuring they look neat and precise.
- Finally, we update the text content of the totalTipElement with the calculated tip and total amounts, formatted as a string.

Step 4: Testing

When the user clicks the "Calculate Tip" button, the calculateTip function is executed. It takes the entered bill amount and service quality, calculates the tip and total amount, and updates the displayed result in the totalTipElement.

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

This project showcases how JavaScript can be used to interact with user input, perform calculations, and dynamically update content on a webpage. It's a great example of a practical application where JavaScript enhances user experience by providing instant feedback on calculations.

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