JavaScript localStorage Counter

Back to JS page

Table of Contents

- JavaScript localStorage Counter
  - Project Overview
  - 1. Create the HTML
    - Example 1
  - 2. Create a Script
    - Example 2
  - 3. Add an Increase Counter Function
    - Example 3
  - 4. Add a Decrease Counter Function
    - Example 4
  - 5. Add a Reset Counter Function
    - Example 5
  - 6. Create a Save Counter Function
    - Example 6
  - 7. Create a Load Counter Function
    - Example 7
  - Exercises & Solutions
    - Example 8
  - Document
  - Reference

Project Overview

In this project, you will build a counter with buttons to:

- **Increase** the counter
- **Decrease** the counter
- **Reset** the counter
- **Save** the counter to local storage
- **Load** the counter from local storage

Project Skills Required:

- JavaScript Variables
- JavaScript Functions

- JavaScript HTML DOM
- JavaScript Events (`onclick` attributes)
- JavaScript `localStorage` (to Save and Restore the counter)

1. Create the HTML

Create the basic HTML layout containing the counter display and buttons:

- Add a `<h2>` header element.
- Add a `<p>` element displaying a number.
- Add 5 `<button>` elements.
- Add an `onclick` attribute to each button to trigger JavaScript functions.

```
<h2>Counter</h2>

<p id="count" style="font-size:40px;">0</p>

<button onclick="increaseCount()">+</button>
<button onclick="decreaseCount()">-</button>
<button onclick="resetCount()">Reset</button>
<button onclick="saveCount()">Save</button>
<button onclick="loadCount()">Load</button>
```

Example 1

Result View Example

2. Create a Script

Create a script that:

- Declares a counter variable: `let count = 0`
- Creates a function `updateCount()` to update the display
- Uses `document.getElementById().innerHTML` to display the counter value

```
<script>
// Declare a counter
let count = 0;

// Function to display the counter
function updateCount() {
  document.getElementById("count").innerHTML = count;
}
</script>
```

Example 2

Result View Example

3. Add an Increase Counter Function

- Create `increaseCount()` to handle incrementing.
- Use `count++` to increase the counter.
- Call `updateCount()` to refresh the display.

```
// Function to increase the counter
function increaseCount() {
  count++;
  updateCount();
}
```

Example 3

Result View Example

4. Add a Decrease Counter Function

- Create `decreaseCount()` to handle decrementing.
- Use `count--` to decrease the counter.
- Call `updateCount()` to refresh the display.

```
// Function to decrease the counter
function decreaseCount() {
  count--;
  updateCount();
}
```

Example 4

Result View Example

5. Add a Reset Counter Function

- Create `resetCount()` to reset.
- Set `count = 0` to reset the counter value.
- Call `updateCount()` to refresh the display.

```
// Function to reset the counter
function resetCount() {
  count = 0;
  updateCount();
}
```

Example 5

Result View Example

6. Create a Save Counter Function

- Create `saveCount()` to save.
- Use `localStorage.setItem()` to store the current count value as text.

```
// Function to save the counter
function saveCount() {
  localStorage.setItem("count", count);
}
```

Example 6

Result View Example

7. Create a Load Counter Function

- Create `loadCount()` to load.
- Use `localStorage.getItem()` to get the saved value.
- Use `Number(saved)` to convert the saved text back into a number.
- Call `updateCount()` to refresh the display.

```
// Function to load the counter
function loadCount() {
  let saved = localStorage.getItem("count");
  if (saved !== null) {
    count = Number(saved);
  }
  updateCount();
}
```

> [!NOTE]
> `localStorage` stores values as text, so we use `Number()` to convert the value back into a number.

Example 7

Result View Example

Exercises & Solutions

Exercise 1

Make the counter start at 10 instead of 0.

Exercise 2

Prevent the counter from going below 0.

Exercise 3

Automatically load the saved counter value when the page opens.

Solutions Code

```
<h2>Counter</h2>

<p id="count" style="font-size:40px;">0</p>

<button onclick="increaseCount()">+</button>
<button onclick="decreaseCount()">-</button>
<button onclick="resetCount()">Reset</button>
<button onclick="saveCount()">Save</button>
<button onclick="loadCount()">Load</button>

<script>
// Declare the counter (starts at 10)
let count = 10;

// Load the counter when the page opens
loadCount();

// Function to display the counter
function updateCount() {
  document.getElementById("count").innerHTML = count;
}

// Function to increase the counter
function increaseCount() {
  count++;
  updateCount();
}

// Function to decrease the counter (prevent going below 0)
function decreaseCount() {
  if (count > 0) {
    count--;
    updateCount();
  }
}

// Function to reset the counter (resets to 10)
function resetCount() {
  count = 10;
  updateCount();
}

// Function to save the counter
function saveCount() {
  localStorage.setItem("count", count);
}

// Function to load the counter
function loadCount() {
  let saved = localStorage.getItem("count");
```

```
  if (saved !== null) {
    count = Number(saved);
  }
  updateCount();
}
</script>
```

Example 8

Result View Example

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript Counter Project