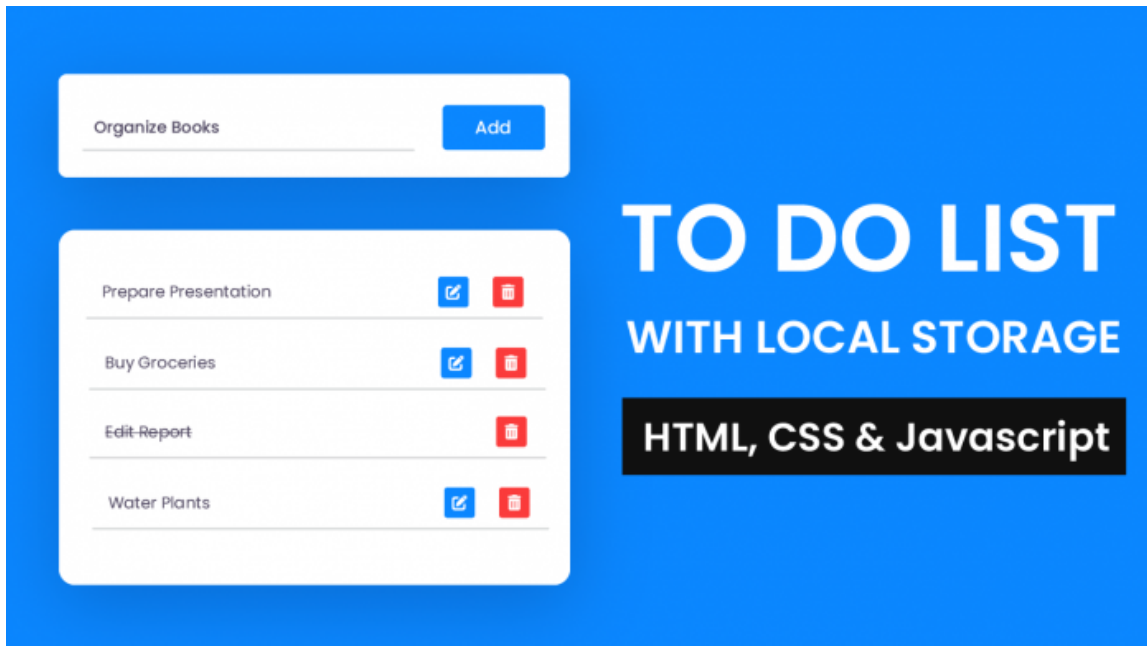


[Home](#) > [Javascript](#) > [To Do App With Local Storage](#)[Javascript](#)[Javascript Projects](#)

To Do App With Local Storage

By **Coding Artist** 18th September 2022

👁 1033 💬 0



Hey everyone. Welcome to this tutorial. In this tutorial, we will learn how to create a to-do list app with local storage. To build this app we need HTML, CSS and Javascript.

This is an intermediate-level javascript project. If you are looking for more projects to improve your javascript skills you can check out [this playlist](#) here. This playlist consists of 100+ javascript projects.

The difficulty level of these projects varies from simple to quite complex ones. Hence these projects are suitable for everyone including javascript beginners to javascript intermediates.

Video Tutorial:

If you are interested to learn by watching a video tutorial rather than reading this blog post you can check out the video down below. Also do not forget to subscribe to [my youtube channel](#) where I post new tips, tricks and tutorials every alternate day.

Along with these, I post multiple choice questions based on HTML, CSS and Javascript that will help you with your interviews.

Project Folder Structure:

Before we begin coding we take a look at the project folder structure. We create a project folder called – 'To Do List With Local Storage'. Within this folder, we have three files. These files are index.html, style.css and script.js. The first file is the HTML document, the next one is the stylesheet and lastly, we have the script file.

HTML:

We begin with the HTML code. First, copy the code below and paste it into your HTML document.

```

1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <meta name="viewport" content="width=device-width, initial-scale=1.0" ,
5.     <title>To Do List With Local Storage</title>
6.     <!-- Font Awesome Icons -->
7.     <link
8.       rel="stylesheet"
9.       href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/fo
10.    />
11.     <!-- Google Fonts -->
12.     <link
13.       href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500&
14.       rel="stylesheet"
15.     />
16.     <!-- Stylesheet -->
17.     <link rel="stylesheet" href="style.css" />
18.   </head>
19.   <body>
20.     <div class="container">
21.       <div id="new-task">
22.         <input type="text" placeholder="Enter The Task Here..." />
23.         <button id="push">Add</button>
24.       </div>
25.       <div id="tasks"></div>
26.     </div>
27.     <!-- Script -->
28.     <script src="script.js"></script>
29.   </body>
30. </html>

```

CSS:

Next, we style this app using CSS. Now copy provided to you below and paste it into your stylesheet.

```

1. * {
2.   padding: 0;
3.   margin: 0;
4.   box-sizing: border-box;
5. }
6. body {
7.   background-color: #0b87ff;
8. }
9. .container {
10.  width: 90%;
11.  max-width: 34em;
12.  position: absolute;
13.  transform: translate(-50%, -50%);
14.  top: 50%;
15.  left: 50%;
16. }
17. #new-task {
18.  position: relative;
19.  background-color: #ffffff;
20.  padding: 1.8em 1.25em;
21.  border-radius: 0.3em;
22.  box-shadow: 0 1.25em 1.8em rgba(1, 24, 48, 0.15);
23.  display: grid;

```

```
24.   grid-template-columns: 9fr 3fr;
25.   gap: 1em;
26. }
27. #new-task input {
28.   font-family: "Poppins", sans-serif;
29.   font-size: 1em;
30.   border: none;
31.   border-bottom: 2px solid #d1d3d4;
32.   padding: 0.8em 0.5em;
33.   color: #111111;
34.   font-weight: 500;
35. }
36. #new-task input:focus {
37.   outline: none;
38.   border-color: #0b87ff;
39. }
40. #new-task button {
41.   font-family: "Poppins", sans-serif;
42.   font-weight: 500;
43.   font-size: 1em;
44.   background-color: #0b87ff;
45.   color: #ffffff;
46.   outline: none;
47.   border: none;
48.   border-radius: 0.3em;
49.   cursor: pointer;
50. }
51. #tasks {
52.   background-color: #ffffff;
53.   position: relative;
54.   padding: 1.8em 1.25em;
55.   margin-top: 3.8em;
56.   width: 100%;
57.   box-shadow: 0 1.25em 1.8em rgba(1, 24, 48, 0.15);
58.   border-radius: 0.6em;
59. }
60. .task {
61.   background-color: #ffffff;
62.   padding: 0.3em 0.6em;
63.   margin-top: 0.6em;
64.   display: flex;
65.   align-items: center;
66.   border-bottom: 2px solid #d1d3d4;
67.   cursor: pointer;
68. }
69. .task span {
70.   font-family: "Poppins", sans-serif;
71.   font-size: 0.9em;
72.   font-weight: 400;
73. }
74. .task button {
75.   color: #ffffff;
76.   padding: 0.8em 0;
77.   width: 2.8em;
78.   border-radius: 0.3em;
79.   border: none;
80.   outline: none;
81.   cursor: pointer;
82. }
83. .delete {
84.   background-color: #fb3b3b;
85. }
86. .edit {
87.   background-color: #0b87ff;
88.   margin-left: auto;
89.   margin-right: 3em;
90. }
```

```

91.   .completed {
92.     text-decoration: line-through;
93.   }

```

Javascript:

Finally, we implement the functionality of this app using javascript. We do this in eight simple steps:

1. Create Initial References
2. Implement Function On Window Load
3. Function To Display Tasks
4. Set Function To Disable Edit Tasks
5. Function To Remove Tasks From Local Storage
6. A Function To Add Tasks To Local Storage
7. Function To Add A New Task
8. Add Functionality To Edit & Delete Buttons

```

1.  //Initial References
2.  const newTaskInput = document.querySelector("#new-task input");
3.  const tasksDiv = document.querySelector("#tasks");
4.  let deleteTasks, editTasks, tasks;
5.  let updateNote = "";
6.  let count;
7.
8.  //Function on window load
9.  window.onload = () => {
10.    updateNote = "";
11.    count = Object.keys(localStorage).length;
12.    displayTasks();
13.  };
14.
15.  //Function to Display The Tasks
16.  const displayTasks = () => {
17.    if (Object.keys(localStorage).length > 0) {
18.      tasksDiv.style.display = "inline-block";
19.    } else {
20.      tasksDiv.style.display = "none";
21.    }
22.
23.    //Clear the tasks
24.    tasksDiv.innerHTML = "";
25.
26.    //Fetch All The Keys in local storage
27.    let tasks = Object.keys(localStorage);
28.    tasks = tasks.sort();
29.
30.    for (let key of tasks) {
31.      let classValue = "";
32.
33.      //Get all values
34.      let value = localStorage.getItem(key);
35.      let taskInnerDiv = document.createElement("div");
36.      taskInnerDiv.classList.add("task");
37.      taskInnerDiv.setAttribute("id", key);
38.      taskInnerDiv.innerHTML = `<span id="taskname">${key.split("_")[1]}</span>`;
39.      //localStorage would store boolean as string so we parse it to boolean
40.      let editButton = document.createElement("button");
41.      editButton.classList.add("edit");
42.      editButton.innerHTML = `<i class="fa-solid fa-pen-to-square"></i>`;

```

```

43.     if (!JSON.parse(value)) {
44.         editButton.style.visibility = "visible";
45.     } else {
46.         editButton.style.visibility = "hidden";
47.         taskInnerDiv.classList.add("completed");
48.     }
49.     taskInnerDiv.appendChild(editButton);
50.     taskInnerDiv.innerHTML += `<button class="delete"><i class="fa-solid fa
51.     tasksDiv.appendChild(taskInnerDiv);
52. }
53.
54. //tasks completed
55. tasks = document.querySelectorAll(".task");
56. tasks.forEach((element, index) => {
57.     element.onclick = () => {
58.         //local storage update
59.         if (element.classList.contains("completed")) {
60.             updateStorage(element.id.split("_")[0], element.innerText, false);
61.         } else {
62.             updateStorage(element.id.split("_")[0], element.innerText, true);
63.         }
64.     };
65. });
66.
67. //Edit Tasks
68. editTasks = document.getElementsByClassName("edit");
69. Array.from(editTasks).forEach((element, index) => {
70.     element.addEventListener("click", (e) => {
71.         //Stop propogation to outer elements (if removed when we click delet
72.         e.stopPropagation();
73.         //disable other edit buttons when one task is being edited
74.         disableButtons(true);
75.         //update input value and remove div
76.         let parent = element.parentElement;
77.         newTaskInput.value = parent.querySelector("#taskname").innerText;
78.         //set updateNote to the task that is being edited
79.         updateNote = parent.id;
80.         //remove task
81.         parent.remove();
82.     });
83. });
84.
85. //Delete Tasks
86. deleteTasks = document.getElementsByClassName("delete");
87. Array.from(deleteTasks).forEach((element, index) => {
88.     element.addEventListener("click", (e) => {
89.         e.stopPropagation();
90.         //Delete from local storage and remove div
91.         let parent = element.parentElement;
92.         removeTask(parent.id);
93.         parent.remove();
94.         count -= 1;
95.     });
96. });
97. };
98.
99. //Disable Edit Button
100. const disableButtons = (bool) => {
101.     let editButtons = document.getElementsByClassName("edit");
102.     Array.from(editButtons).forEach((element) => {
103.         element.disabled = bool;
104.     });
105. };
106.
107. //Remove Task from local storage
108. const removeTask = (taskValue) => {
109.     localStorage.removeItem(taskValue);

```

```
110.     displayTasks();
111.   };
112.
113.   //Add tasks to local storage
114.   const updateStorage = (index, taskValue, completed) => {
115.     localStorage.setItem(`${index}_${taskValue}`, completed);
116.     displayTasks();
117.   };
118.
119.   //Function To Add New Task
120.   document.querySelector("#push").addEventListener("click", () => {
121.     //Enable the edit button
122.     disableButtons(false);
123.     if (newTaskInput.value.length == 0) {
124.       alert("Please Enter A Task");
125.     } else {
126.       //Store locally and display from local storage
127.       if (updateNote == "") {
128.         //new task
129.         updateStorage(count, newTaskInput.value, false);
130.       } else {
131.         //update task
132.         let existingCount = updateNote.split("_")[0];
133.         removeTask(updateNote);
134.         updateStorage(existingCount, newTaskInput.value, false);
135.         updateNote = "";
136.       }
137.       count += 1;
138.       newTaskInput.value = "";
139.     }
140.   });
```

source code by clicking on the 'Download Code' button below. Also if you have any queries, suggestions or feedback you can comment below.
Happy Coding!

[Download Code](#)