



TP 08

WEB DESIGN

LECTURER : HOK TIN
STUDENT : SENG ENGHAV
ID E20200159




LINK GITHUB

<https://github.com/Enghav/TP8>

TP08.1

To-Do List



GIC	Eno	2000-01-01	<input type="button" value="Delete"/>
-----	-----	------------	---------------------------------------

Application

Manifest

Service Workers

Storage

Storage

Local Storage

http://127.0.0.1:5500/

Session Storage

IndexedDB

Web SQL

Cookies

Private State Tokens

Interest Groups

Shared Storage

Cache Storage

Background Services

Back/forward cache

Background Fetch

Background Sync

Notifications

Payment Handler

Periodic Background Sync

Push Messaging

Reporting API

Frames

top

Filter

Key	Value
tasks	[{"title": "GIC", "assignee": "Eno", "date": "2000-01-01"}]

[{"title": "GIC", "assignee": "Eno", "date": "2000-01-01"}]

0: {"title": "GIC", "assignee": "Eno", "date": "2000-01-01"}

```

<!DOCTYPE html>
<html Lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>To-Do List local</title>
    <script src="https://cdn.tailwindcss.com"></script>
  </head>
  <body>
    <div class="container mx-auto p-4">
      <h1 class="text-2xl font-bold mb-4 text-center">To-Do List</h1>
      <form id="taskForm" class="mb-4">
        <div
          class="flex items-center w-[700px] justify-between ml-8 bg-slate-500 p-5 rounded-[10px]"
        >
          <input
            id="taskTitle"
            type="text"
            placeholder="Task Title"
            class="border rounded-l py-2 px-4 w-[150px]"
            required/>
          <input
            id="assignee"
            type="text"
            placeholder="Assignee"
            class="border py-2 px-4 w-[150px]"
            required />
          <input
            id="taskDate"
            type="date"
            class="border rounded-r py-2 px-4 w-[150px]"
            required/>
          <button
            type="submit"
            class="bg-blue-500 hover:bg-blue-700 text-white py-2 px-4 ml-2 rounded">
            Add Task
          </button>
        </div>
      </form>
      <ul id="taskList" class="list-disc pl-8"></ul>
    </div>
    <script src="local.js"></script>
  </body>
</html>

```

```

const taskForm = document.getElementById("taskForm");
const taskList = document.getElementById("taskList");
const taskTitleInput = document.getElementById("taskTitle");
const assigneeInput = document.getElementById("assignee");
const taskDateInput = document.getElementById("taskDate");
let tasks = JSON.parse(localStorage.getItem("tasks")) || [];
function renderTasks() {
  taskList.innerHTML = "";

  tasks.forEach((task, index) => {
    const listItem = document.createElement("tasks");
    listItem.innerHTML = `
      <div class="flex w-[700px] justify-between border-2 p-2 rounded-[10px] text-slate-700 m-2 ml-0">
        <span>${task.title}</span>
        <span>${task.assignee}</span>
        <span>${task.date}</span>
        <button class="bg-red-500 hover:bg-red-700 text-white py-1 px-2 ml-2 rounded" onclick="deleteTask(${index})">Delete</button>
      </div>
    `;
    taskList.appendChild(listItem);
  });
}
function addTask(event) {
  event.preventDefault();

  const title = taskTitleInput.value;
  const assignee = assigneeInput.value;
  const date = taskDateInput.value;

  const task = { title, assignee, date };
  tasks.push(task);
  localStorage.setItem("tasks", JSON.stringify(tasks));
  taskTitleInput.value = "";
  assigneeInput.value = "";
  taskDateInput.value = "";
  renderTasks();
}

function deleteTask(index) {
  tasks.splice(index, 1);

  localStorage.setItem("tasks", JSON.stringify(tasks));

  renderTasks();
}

taskForm.addEventListener("submit", addTask);

renderTasks();

```

TP08.2

To-Do List

Task Title

Assignee

mm/dd/yyyy

Add Task

Gic

Jing Jork

2023-07-21

Delete

Manifest

Service Workers

Storage

Storage

Local Storage

Session Storage

http://127.0.0.1:5500/

IndexedDB

Web SQL

Cookies

Private State Tokens

Interest Groups

Shared Storage

Cache Storage

Background Services

Back/forward cache

Background Fetch

Background Sync

Notifications

Payment Handler

Periodic Background Sync

Push Messaging

Reporting API

Key	Value
IsThisFirstTime_Log_From_LiveServer	true
tasks	[{"title":"Gic","assignee":"Jing Jork","da...
▼ [{"title": "Gic", assignee: "Jing Jork", date: "2023-07-21"}] ▶ 0: {title: "Gic", assignee: "Jing Jork", date: "2023-07-21"}	

```
<!DOCTYPE html>
<html Lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>To-Do List session</title>
    <script src="https://cdn.tailwindcss.com"></script>
  </head>
  <body class="bg-">
    <div class="container mx-auto p-4">
      <h1 class="text-2xl font-bold mb-4 text-center">To-Do List</h1>
      <form id="taskForm" class="mb-4">
        <div
          class="flex items-center w-[700px] justify-between ml-8 bg-slate-500 p-5 rounded-[10px]" >
          <input
            id="taskTitle"
            type="text"
            placeholder="Task Title"
            class="border rounded-l py-2 px-4 w-[150px]"
            required/>
          <input
            id="assignee"
            type="text"
            placeholder="Assignee"
            class="border py-2 px-4 w-[150px]"
            required/>
          <input
            id="taskDate"
            type="date"
            class="border rounded-r py-2 px-4 w-[150px]"
            required/>
          <button
            type="submit"
            class="bg-blue-500 hover:bg-blue-700 text-white py-2 px-4 ml-2 rounded">
            Add Task
          </button>
        </div>
      </form>
      <ul id="taskList" class="list-disc pl-8"></ul>
    </div>
    <script src="session.js"></script>
  </body>
</html>
```

```
const taskForm = document.getElementById("taskForm");
const taskList = document.getElementById("taskList");
const taskTitleInput = document.getElementById("taskTitle");
const assigneeInput = document.getElementById("assignee");
const taskDateInput = document.getElementById("taskDate");
let tasks = JSON.parse(sessionStorage.getItem("tasks")) || [];
function renderTasks() {
  taskList.innerHTML = "";

  tasks.forEach((task, index) => {
    const listItem = document.createElement("tasks");
    listItem.innerHTML = `
      <div class="flex w-[700px] justify-between border-2 p-2 rounded-[10px] text-slate-700 mb-[4px]">
        <span>${task.title}</span>
        <span>${task.assignee}</span>
        <span>${task.date}</span>
        <button class="bg-red-500 hover:bg-red-700 text-white py-1 px-2 ml-2 rounded" onclick="deleteTask(${index})">Delete</button>
      </div>
    `;
    taskList.appendChild(listItem);
  });
}
function addTask(event) {
  event.preventDefault();

  const title = taskTitleInput.value;
  const assignee = assigneeInput.value;
  const date = taskDateInput.value;

  const task = { title, assignee, date };
  tasks.push(task);
  sessionStorage.setItem("tasks", JSON.stringify(tasks));
  taskTitleInput.value = "";
  assigneeInput.value = "";
  taskDateInput.value = "";
  renderTasks();
}

function deleteTask(index) {
  tasks.splice(index, 1);

  sessionStorage.setItem("tasks", JSON.stringify(tasks));

  renderTasks();
}

taskForm.addEventListener("submit", addTask);


renderTasks();
```

TP08.3

Cookie Management

Key

Value

mm/dd/yyyy 

Add Task

Gic

Eno

2023-07-06

Delete

Gic

Jing Jork

2023-07-26

Delete

[illegible]



```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title> cookie management</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
```

```
<body >
  <div class="container mx-auto p-4">
    <h1 class="text-2xl font-bold mb-4 text-center">Cookie Management</h1>
    <form id="taskForm" class="mb-4">
      <div class="flex items-center w-[700px] justify-between ml-8 bg-slate-500 p-5 rounded-[10px]">
        <input id="Key" type="text" placeholder="Key" class="border rounded-l py-2 px-4 w-[150px]"
          required>
        <input id="Value" type="text" placeholder="Value" class="border py-2 px-4 w-[150px]" required>
        <input id="ExpiredDate" type="date" class="border rounded-r py-2 px-4 w-[150px]" required >
        <button type="submit"
          class="bg-blue-500 hover:bg-blue-700 text-white py-2 px-4 ml-2 rounded">Add Task</button>
      </div>
    </form>
    <ul id="taskList" class="list-disc pl-8"></ul>
  </div>
```

```

  <script src="cookie.js">
</script>
</body>
```

```
</html>
```



```
const taskForm = document.getElementById("taskForm");
const taskList = document.getElementById("taskList");
const taskTitleInput = document.getElementById("Key");
const assigneeInput = document.getElementById("Value");
const taskDateInput = document.getElementById("ExpiredDate");
let tasks = [];
```

```
// Function to set a cookie
function setCookie(name, value, days) {
  const expires = new Date();
  expires.setTime(expires.getTime() + days * 24 * 60 * 60 * 1000);
  document.cookie = `${name}=${value};expires=${expires.toUTCString()};path=/`;
}
```

```
// Function to get a cookie
function getCookie(name) {
  const cookieArr = document.cookie.split(";");
  for (let i = 0; i < cookieArr.length; i++) {
    const cookiePair = cookieArr[i].split("=");
    if (name === cookiePair[0].trim()) {
      return decodeURIComponent(cookiePair[1]);
    }
  }
  return null;
}
```

```
function renderTasks() {
  taskList.innerHTML = "";
```

```
  tasks.forEach((task, index) => {
    const listItem = document.createElement("tasks");
    listItem.innerHTML = `
      <div class="flex w-[700px] justify-between border-2 p-2 rounded-[10px] text-slate-700 m-2 ml-0">
        <span>${task.key}</span>
        <span>${task.value}</span>
        <span>${task.exdate}</span>
        <button class="bg-red-500 hover:bg-red-700 text-white py-1 px-2 ml-2 rounded" onclick="deleteTask(${index})">Delete</button>
      </div>
    `;
    taskList.appendChild(listItem);
  });
}
```

```
function addTask(event) {
  event.preventDefault();
```

```

  const key = taskTitleInput.value;
  const value = assigneeInput.value;
  const exdate = taskDateInput.value;
```

```

  const cookie = { key, value, exdate };
  tasks.push(cookie);
  setCookie("tasks", JSON.stringify(tasks), 30); // Store the tasks in a cookie for 30 days
  taskTitleInput.value = "";
  assigneeInput.value = "";
  taskDateInput.value = "";
  renderTasks();
}
```

```
function deleteTask(index) {
  tasks.splice(index, 1);
  setCookie("tasks", JSON.stringify(tasks), 30); // Update the tasks cookie after deletion
  renderTasks();
}
```

```
taskForm.addEventListener("submit", addTask);
```

```
// Load tasks from the cookie when the page loads
```

```
const storedTasks = getCookie("tasks");
if (storedTasks) {
  tasks = JSON.parse(storedTasks);
  renderTasks();
}
```


WEB DESIGN

THE END