Lab — JavaScript Review: Task 2  
ToDo List with Array (Plain JavaScript)

# What you’ll build

A ToDo app that keeps tasks in a JavaScript array, renders the list from that array, and re-renders whenever you add items.

# Key concepts

* State: the array is the single source of truth for tasks.
* Rendering from state: render() clears the <ul> and rebuilds its <li> children from tasks[].
* Events: clicking Add pushes to tasks[] and calls render(); Enter key does the same.

# Step-by-step

1. Make the same HTML skeleton as Task 1: input, button, and an empty <ul id="todo-list">.
2. Declare const tasks = ["Task 1","Task 2","Task 3"];
3. Write function render() that sets list.innerHTML = "" and then loops tasks to create and append <li> items.
4. In the click handler: read input.value.trim(); if present, tasks.push(text); clear input; call render().
5. Add the Enter-key handler to call the button click.
6. Call render() once at the bottom to paint the initial array contents.

# Complete example (copy & run)

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>ToDo List — Plain JS with Array (Task 2)</title>

<style>

body { font-family: system-ui, -apple-system, Segoe UI, Roboto, sans-serif; max-width: 720px; margin: 40px auto; padding: 0 16px; }

header { display: flex; gap: 8px; }

input[type=text] { flex: 1; padding: 8px; }

button { padding: 8px 12px; cursor: pointer; }

ul { padding-left: 20px; }

li { padding: 6px 0; }

</style>

</head>

<body>

<h1>ToDo List — Plain JS with Array (Task 2)</h1>

<p>Items are stored in a JavaScript array and rendered from it.</p>

<header>

<input id="new-task" type="text" placeholder="New task..." />

<button id="add-btn">Add task</button>

</header>

<ul id="todo-list"></ul>

<script>

const input = document.getElementById('new-task');

const button = document.getElementById('add-btn');

const list = document.getElementById('todo-list');

const tasks = ["Task 1", "Task 2", "Task 3"];

function render() {

list.innerHTML = "";

for (const t of tasks) {

const li = document.createElement('li');

li.textContent = t;

list.appendChild(li);

}

}

button.addEventListener('click', () => {

const text = input.value.trim();

if (!text) return;

tasks.push(text);

input.value = "";

input.focus();

render();

});

input.addEventListener('keyup', (e) => {

if (e.key === 'Enter') button.click();

});

render();

</script>

</body>

</html>

# Common enhancements

* Clear all button: set tasks.length = 0 and render().
* Reset demo button: tasks.splice(0, tasks.length, 'Task 1','Task 2','Task 3'); render().
* Delete/edit per item (future exercise).

# Assessment checklist

* Three tasks pre-populated in the array.
* Adding a task pushes to the array and re-renders the list.
* Blank entries are ignored.
* render() is the single place that builds the UI from state.