|  |
| --- |
| HTML |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Todo List</title>      <script type="text/javascript" src="./assets/js/lib/jquery-3.5.0.min.js"></script>      <link rel="stylesheet" href="./assets/css/todos2.css">      <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=swap" rel="stylesheet">      <link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet"  crossorigin="anonymous">  </head>  <body>      <h1><span class="styling">TODO</span>List</h1>      <div class="input\_div">          <input type="text" class="input" placeholder="What Do You Want to Do...">          <button class="addButton"><i class="fa fa-plus"></i> </button>      </div>      <div class="container">         <!-- <div class="item">              <input type="text" class="item\_input"  disabled>              <button class="editButton">Edit</button>              <button class="removeButton">Remove</button>          </div> -->      </div>      <script type="text/javascript" src="./assets/js/todos3.js"></script>  </body>  </html> |
| CSS |
| \* {      padding: 0;      margin: 0;      box-sizing: border-box;}  body {      height: 100%;      background-color: #111;}  h1 {      font-size: 3rem;      font-weight: 50;      margin: 1rem 0 3rem;      text-align: center;      color: #fff;      font-family: 'Roboto', sans-serif;  }  .styling {      font-weight: 800;      color: rgb(255, 81, 0);}  .input\_div {      display: flex;      justify-content: center;      align-items: center;      text-align: center;}  .input\_div .input {      padding: 0.5rem 1rem;      height: 50px;      outline: none;      border: none;      background-color: #fff;      width: 450px;      font-size: 1.15rem;      margin: 0.25rem;      border-radius: 10px;}  .addButton {      height: 50px;      width: 50px;      border-radius: 10px;      outline: none;      border: none;      color: #fff;      margin: 0.25rem;      background-color:  rgb(255, 81, 0);      cursor: pointer;}  .addButton:hover {      outline: 0.7;}  .container {      display: flex;      flex-direction: column;      justify-content: center;      align-items: center;      text-align: center;      margin-top: 2rem;}  .item {      padding: 0.5rem;      margin-bottom: 1.5rem;      border-bottom: 4px solid #fff;}  .item\_input {      background: none;      outline: none;      color: #fff;      border: none;      width: 350px;      font-size: 1.4rem;}  .editButton {      font-size: 1.4rem;      margin: 0 0.5rem;      font-family: 'Roboto', sans-serif;      background: none;      outline: none;      color: rgb(255, 81, 0);      border: none;      cursor: pointer;}  .removeButton {      font-size: 1.4rem;      margin: 0 0.5rem;      font-family: 'Roboto', sans-serif;      background: none;      outline: none;      color: rgb(255, 0, 0);      border: none;      cursor: pointer;} |

|  |
| --- |
| JavaScript |
| const addButton = document.querySelector('.addButton');  var input = document.querySelector('.input');  const container = document.querySelector('.container');  //we create class with prototype value 'item'  class item {      constructor (itemName) {          //create the item div method          this.createDiv(itemName);      }      createDiv(itemName) {          //input for create          let input = document.createElement('input');          //create value, disable(true or false), add classList and value('text')          input.value = itemName;          input.disabled = true;          input.classList.add('item\_input');          input.type = "text";          //create method for div and add class itemBox          let itemBox = document.createElement('div');          //itemBox -> className -> item          itemBox.classList.add('item');          //create a method for button RESET          let editButton = document.createElement('button');          //add InnerHTML with value EDIT          editButton.innerHTML = "EDIT";          //add className -> removeButton          editButton.classList.add('editButton');          //do the sane for button REMOVE          //create a method for button REMOVE          let removeButton = document.createElement('button');          //add InnerHTML with value EDIT          removeButton.innerHTML = "REMOVE";          //add className -> removeButton          removeButton.classList.add('removeButton');          //add method to call all child in var -> container          container.appendChild(itemBox);          //call className -> input          itemBox.appendChild(input);          //call className -> editButton          itemBox.appendChild(editButton);          //call className -> removeButton          itemBox.appendChild(removeButton);          //add callback function to use input type          editButton.addEventListener('click', () => this.edit(input));          //add callback function to use button REMOVE          removeButton.addEventListener('click', () => this.remove(itemBox));      }      edit(input) {          input.disabled = !input.disabled;      }      remove(item) {          container.removeChild(item);      }  }  function check () {      if(input.value != "") {          new item(input.value);          input.value = "";      }  }  addButton.addEventListener('click', check);  window.addEventListener('click', check);  window.addEventListener('keydown', (e) => {      if(e.which == 13) {          check();      }  }) |