|  |
| --- |
| let root = document.getElementById("root"); |
|  |  |
|  |  |
|  | class todoList{ |
|  | constructor(place, title = "to-do list"){ |
|  |  |
|  | this.place = place; |
|  | this.title = title; |
|  | this.cardArray = []; |
|  |  |
|  | this.render(); |
|  | } |
|  |  |
|  | addToDo(){ |
|  | let text = this.input.value; |
|  | this.cardArray.push(new Card(text, this.div, this)); |
|  | } |
|  |  |
|  | render(){ |
|  | this.createToDoListElement(); |
|  | this.place.append(this.todoListElement); |
|  | } |
|  |  |
|  | createToDoListElement(){ |
|  | //Create elements |
|  | this.h2 = document.createElement('h2'); |
|  | this.h2.innerText = this.title; |
|  | this.input = document.createElement('input'); |
|  | this.input.classList.add("comment"); |
|  | this.button = document.createElement('button'); |
|  | this.button.innerText = 'Add'; |
|  | this.button.classList.add("btn-save"); |
|  | this.button.id = "to-do-list-button"; |
|  | this.div = document.createElement('div'); |
|  | this.todoListElement = document.createElement('div'); |
|  |  |
|  | //Add Event listener |
|  | this.button.addEventListener('click', ()=>{ |
|  | if(this.input.value != ""){ |
|  | this.addToDo.call(this); |
|  | this.input.value = ""; |
|  | } |
|  | }); |
|  |  |
|  | //Append elements to the to-do list element |
|  | this.todoListElement.append(this.h2); |
|  | this.todoListElement.append(this.input); |
|  | this.todoListElement.append(this.button); |
|  | this.todoListElement.append(this.div); |
|  | this.todoListElement.classList.add("todoList"); |
|  | } |
|  | } |
|  |  |
|  |  |
|  | class Card{ |
|  | constructor(text, place, todoList){ |
|  |  |
|  | this.place = place; |
|  | this.todoList = todoList; |
|  | this.state = { |
|  | text: text, |
|  | description: "Click to write a description...", |
|  | comments: [] |
|  | } |
|  | this.render(); |
|  | } |
|  |  |
|  | render(){ |
|  | this.card = document.createElement('div'); |
|  | this.card.classList.add("card"); |
|  | this.card.addEventListener('click', (e)=>{ |
|  | if(e.target != this.deleteButton){ |
|  | this.showMenu.call(this); |
|  | } |
|  | }); |
|  |  |
|  | this.p = document.createElement('p'); |
|  | this.p.innerText = this.state.text; |
|  |  |
|  | this.deleteButton = document.createElement('button'); |
|  | this.deleteButton.innerText = "X"; |
|  | this.deleteButton.addEventListener('click', ()=>{ |
|  | this.deleteCard.call(this); |
|  | }); |
|  |  |
|  | this.card.append(this.p); |
|  | this.card.append(this.deleteButton); |
|  |  |
|  | this.place.append(this.card); |
|  | } |
|  |  |
|  | deleteCard(){ |
|  | this.card.remove(); |
|  | let i = this.todoList.cardArray.indexOf(this); |
|  | this.todoList.cardArray.splice(i,1); |
|  | } |
|  |  |
|  | showMenu(){ |
|  |  |
|  | //Create elements |
|  | this.menu = document.createElement("div"); |
|  | this.menuContainer = document.createElement("div"); |
|  | this.menuTitle = document.createElement("div"); |
|  | this.menuDescription = document.createElement("div"); |
|  | this.commentsInput = document.createElement("input"); |
|  | this.commentsButton = document.createElement('button'); |
|  | this.menuComments = document.createElement("div"); |
|  |  |
|  |  |
|  | //Add class names |
|  | this.menu.className = "menu"; |
|  | this.menuContainer.className = "menuContainer"; |
|  | this.menuTitle.className = "menuTitle"; |
|  | this.menuDescription.className = "menuDescription"; |
|  | this.menuComments.className = "menuComments"; |
|  | this.commentsInput.className = "commentsInput comment"; |
|  | this.commentsButton.className = "commentsButton btn-save"; |
|  |  |
|  | //Add inner Text |
|  | this.commentsButton.innerText = "Add"; |
|  | this.commentsInput.placeholder = "Write a comment..."; |
|  |  |
|  | //Event listeners |
|  | this.menuContainer.addEventListener('click', (e)=>{ |
|  | console.log(e.target); |
|  | if(e.target.classList.contains("menuContainer")){ |
|  | this.menuContainer.remove(); |
|  | } |
|  | }); |
|  |  |
|  | this.commentsButton.addEventListener('click', ()=>{ |
|  | if(this.commentsInput.value != ""){ |
|  | this.state.comments.push(this.commentsInput.value); |
|  | this.renderComments(); |
|  | this.commentsInput.value = ""; |
|  | } |
|  | }) |
|  |  |
|  | //Append |
|  | this.menu.append(this.menuTitle); |
|  | this.menu.append(this.menuDescription); |
|  | this.menu.append(this.commentsInput); |
|  | this.menu.append(this.commentsButton); |
|  | this.menu.append(this.menuComments); |
|  | this.menuContainer.append(this.menu); |
|  | root.append(this.menuContainer); |
|  |  |
|  | this.editableDescription = new EditableText(this.state.description, this.menuDescription, this, "description", "textarea"); |
|  | this.editableTitle = new EditableText(this.state.text, this.menuTitle, this, "text", "input"); |
|  |  |
|  | this.renderComments(); |
|  | } |
|  |  |
|  | renderComments(){ |
|  |  |
|  | let currentCommentsDOM = Array.from(this.menuComments.childNodes); |
|  |  |
|  | currentCommentsDOM.forEach(commentDOM =>{ |
|  | commentDOM.remove(); |
|  | }); |
|  |  |
|  | this.state.comments.forEach(comment =>{ |
|  | new Comment(comment, this.menuComments, this); |
|  | }); |
|  | } |
|  | } |
|  |  |
|  | class EditableText{ |
|  | constructor(text, place, card, property, typeOfInput){ |
|  | this.text = text; |
|  | this.place = place; |
|  | this.card = card; |
|  | this.property = property; |
|  | this.typeOfInput = typeOfInput; |
|  | this.render(); |
|  | } |
|  |  |
|  | render(){ |
|  | this.div = document.createElement("div"); |
|  | this.p = document.createElement("p"); |
|  |  |
|  | this.p.innerText = this.text; |
|  |  |
|  | this.p.addEventListener('click', ()=>{ |
|  | this.showEditableTextArea.call(this); |
|  | }); |
|  |  |
|  | this.div.append(this.p); |
|  | this.place.append(this.div); |
|  | } |
|  |  |
|  | showEditableTextArea(){ |
|  | let oldText = this.text; |
|  |  |
|  | this.input = document.createElement(this.typeOfInput); |
|  | this.saveButton = document.createElement("button"); |
|  |  |
|  | this.p.remove(); |
|  | this.input.value = oldText; |
|  | this.saveButton.innerText = "Save"; |
|  | this.saveButton.className = "btn-save"; |
|  | this.input.classList.add("comment"); |
|  |  |
|  | this.saveButton.addEventListener('click', ()=>{ |
|  | this.text = this.input.value; |
|  | this.card.state[this.property] = this.input.value; |
|  | if(this.property == "text"){ |
|  | this.card.p.innerText = this.input.value; |
|  | } |
|  | this.div.remove(); |
|  | this.render(); |
|  | }); |
|  |  |
|  | function clickSaveButton(event, object){ |
|  | // Number 13 is the "Enter" key on the keyboard |
|  | if (event.keyCode === 13) { |
|  | // Cancel the default action, if needed |
|  | event.preventDefault(); |
|  | // Trigger the button element with a click |
|  | object.saveButton.click(); |
|  | } |
|  | } |
|  |  |
|  | this.input.addEventListener("keyup", (e)=>{ |
|  | if(this.typeOfInput == "input"){ |
|  | clickSaveButton(e, this); |
|  | } |
|  | }); |
|  |  |
|  | this.div.append(this.input); |
|  |  |
|  | if(this.typeOfInput == "textarea"){ |
|  | this.div.append(this.saveButton); |
|  | } |
|  |  |
|  | this.input.select(); |
|  | } |
|  |  |
|  | } |
|  |  |
|  | class Comment{ |
|  | constructor(text, place, card){ |
|  | this.text = text; |
|  | this.place = place; |
|  | this.card = card; |
|  | this.render(); |
|  | } |
|  |  |
|  | render(){ |
|  | this.div = document.createElement('div'); |
|  | this.div.className = "comment"; |
|  | this.div.innerText = this.text; |
|  |  |
|  | this.place.append(this.div); |
|  | } |
|  | } |
|  |  |
|  |  |
|  |  |
|  | //-------------main------------ |
|  |  |
|  | let addTodoListInput = document.getElementById("addTodoListInput"); |
|  | let addTodoListButton = document.getElementById("addTodoListButton"); |
|  |  |
|  | addTodoListButton.addEventListener('click',()=>{ |
|  | if ( addTodoListInput.value.trim() != ""){ |
|  | new todoList(root, addTodoListInput.value); |
|  | addTodoListInput.value = ""; |
|  | } |
|  | }); |
|  |  |
|  |  |
|  |  |
|  | let todoList1 = new todoList(root); |
|  | let todoList2 = new todoList(root); |
|  | let todoList3 = new todoList(root); |
|  |  |
|  |  |
|  |  |
|  | todoList1.input.value = "asdasds"; |
|  | todoList1.addToDo(); |

|  |
| --- |
| \*{ |
|  | padding: 0; |
|  | margin: 0; |
|  | } |
|  |  |
|  | body{ |
|  | /\*background-color: rgb(30, 30, 95);\*/ |
|  | background-color: rgb(43, 43, 83); |
|  | font-family: "Open Sans"; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | button{ |
|  | outline: none; |
|  | cursor: pointer; |
|  | } |
|  |  |
|  | #addTodoListDiv{ |
|  | margin-left: 2em; |
|  | margin-top: 4em; |
|  | } |
|  |  |
|  | #addTodoListDiv button{ |
|  | margin-left: 0.5em; |
|  | } |
|  |  |
|  |  |
|  | #root{ |
|  | margin: 1em; |
|  | /\*border: 1px solid grey;\*/ |
|  | /\*display: flex;\*/ |
|  | min-width: 1200px; |
|  | } |
|  |  |
|  |  |
|  | .todoList{ |
|  | /\*border: 1px solid red;\*/ |
|  | border-radius: 4px; |
|  | min-height: 100px; |
|  | background: rgb(235, 235, 235); |
|  | padding: 0.7em; |
|  | margin: 1em 0 1em 1em; |
|  |  |
|  | float: left; |
|  | } |
|  |  |
|  | .todoList h2{ |
|  | font-size: 1em; |
|  | margin-bottom: 0.5em; |
|  | } |
|  |  |
|  | #to-do-list-button{ |
|  | margin-left: 0.5em; |
|  | } |
|  |  |
|  | .card{ |
|  | /\*border: 1px solid blue;\*/ |
|  | border-radius: 4px; |
|  | border-bottom: rgb(180, 180, 180) 1px solid; |
|  | background: white; |
|  | margin: 0.5em 0 0 0; |
|  | padding: 0.5em; |
|  | font-size: 0.9em; |
|  | position: relative; |
|  |  |
|  | display: flex; |
|  | justify-content: space-between; |
|  |  |
|  | cursor: pointer; |
|  | } |
|  |  |
|  | .card button{ |
|  | visibility: hidden; |
|  | height: max-content; |
|  |  |
|  | background: none; |
|  | border: none; |
|  | padding: 0.3em; |
|  |  |
|  | } |
|  |  |
|  |  |
|  | .card:hover button{ |
|  | visibility: visible; |
|  | } |
|  |  |
|  | .card button:hover{ |
|  | background-color: rgb(235, 235, 235); |
|  | border-radius: 4px; |
|  | cursor: pointer; |
|  | } |
|  |  |
|  |  |
|  | .menuContainer{ |
|  | display: flex; |
|  |  |
|  | position: absolute; |
|  | top: 0; |
|  | left: 0; |
|  | background-color: rgba(0, 0, 0, 0.8); |
|  | width: 100%; |
|  | height: 100%; |
|  | } |
|  |  |
|  | .menu{ |
|  | top: 0; |
|  | left: 0; |
|  | background-color: rgb(235, 235, 235); |
|  | width: 500px; |
|  | min-height: 300px; |
|  |  |
|  | margin: auto; |
|  |  |
|  | padding: 1em; |
|  | border-radius: 4px; |
|  |  |
|  | } |
|  |  |
|  | .menuTitle{ |
|  | font-weight: bold; |
|  | font-size: 1.5em; |
|  | margin-bottom: 1em; |
|  | } |
|  |  |
|  | .menuDescription{ |
|  | margin-bottom: 2em; |
|  | line-height: 1.5em; |
|  | } |
|  |  |
|  | .menuDescription textarea{ |
|  | width: 100%; |
|  | height: 5em; |
|  | padding: 0.5em; |
|  | font-size: 1.1em; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | .comment{ |
|  | background-color: white; |
|  | border: 1px solid #ccc; |
|  | border-radius: 4px; |
|  | margin-top: 0.5em; |
|  | padding: 0.5em; |
|  | font-size: 0.8em; |
|  | } |
|  |  |
|  | .commentsInput{ |
|  | margin-right: 0.5em; |
|  | } |
|  |  |
|  | .btn-save{ |
|  | background-color: #5aac44; |
|  | color: white; |
|  | border: none; |
|  | border-radius: 4px; |
|  |  |
|  | padding: 0.5em; |
|  | margin-top: 0.5em; |
|  |  |
|  | } |

|  |
| --- |
| <html> |
|  |  |
|  | <head> |
|  | <link rel="stylesheet" href="style.css"></link> |
|  | </head> |
|  |  |
|  |  |
|  | <body> |
|  |  |
|  | <div id="addTodoListDiv"> |
|  | <input id="addTodoListInput" class="comment"> |
|  | <button id="addTodoListButton" class="btn-save">Add new to-do list</button> |
|  | </div> |
|  |  |
|  | <div id="root"></div> |
|  |  |
|  | </body> |
|  |  |
|  | <script src="script.js"></script> |
|  |  |
|  | </html> |