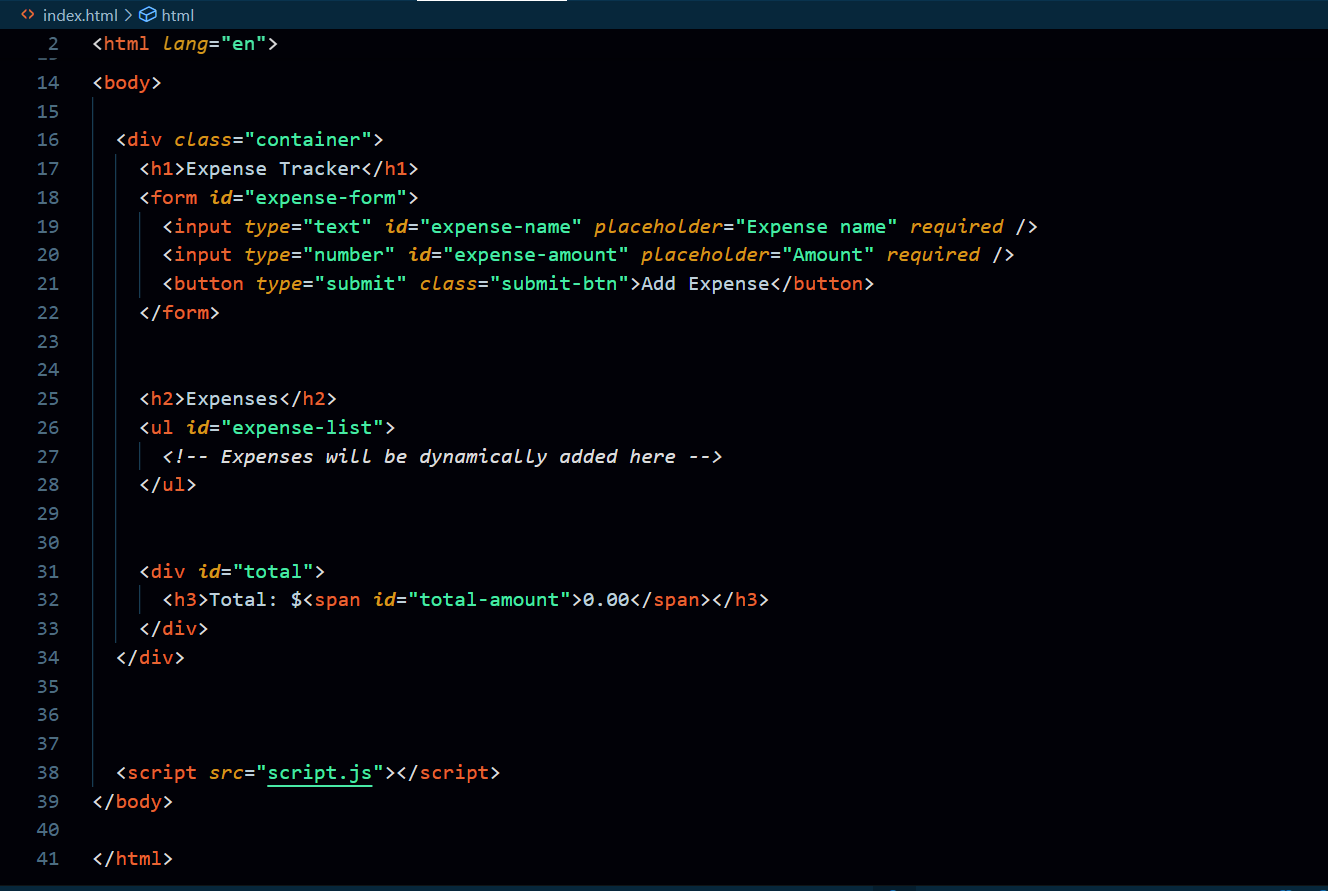
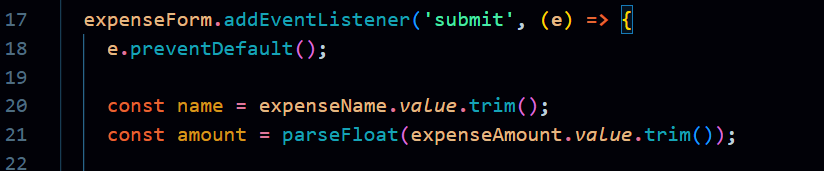
1. **Create HTML CSS files with proper given ID**

****

1. **Create JS file(script.js)**
2. **document.eventListener(DOMContentLoaded, arrow function)**
3. **get all IDs in script.js with proper variable names**

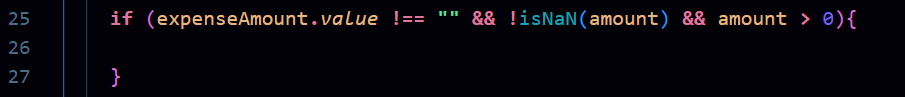
****

1. **Declare expense as an empty Array + Declare totalAmount = calculateTotal();**
2. **Add event listener on form, prevent default, access & store expense Name and expense Amount in a variable.** *(number from input arrives in* ***string*** *format 🡺 convert it to* ***number*** *or* ***float****)*

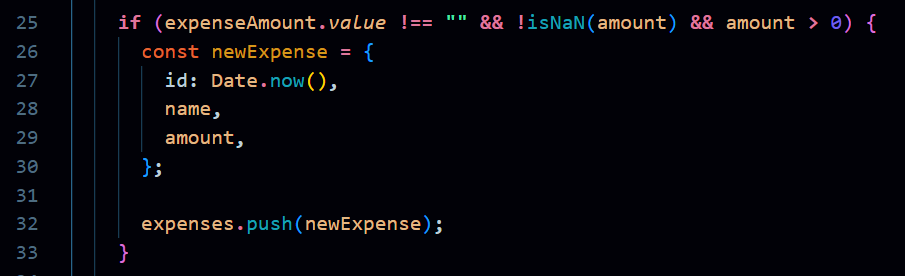
****

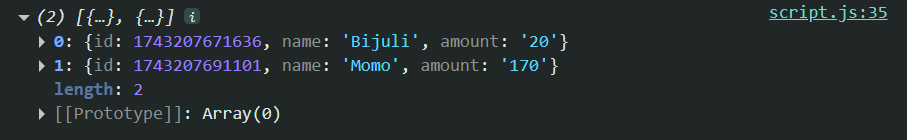
1. **Check on name-input & amount-input (if conditional)**

*(expense not blank, amount not NaN & amont > 0)*

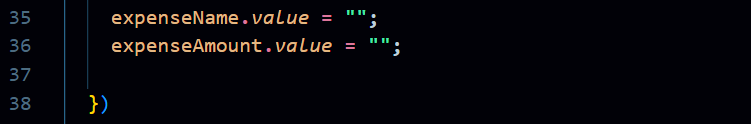
**

1. **Create an *newExpense Object* inside if-conditional & push on expense array**

****

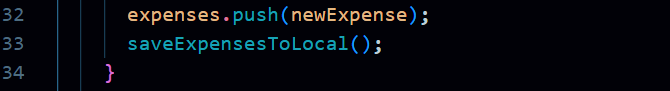
****

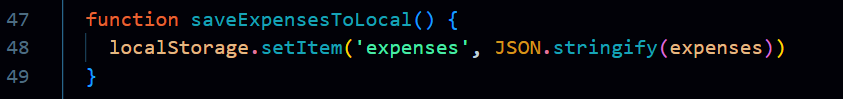
1. **Clear input fields 🡺 at the end of expenseForm add event listener**

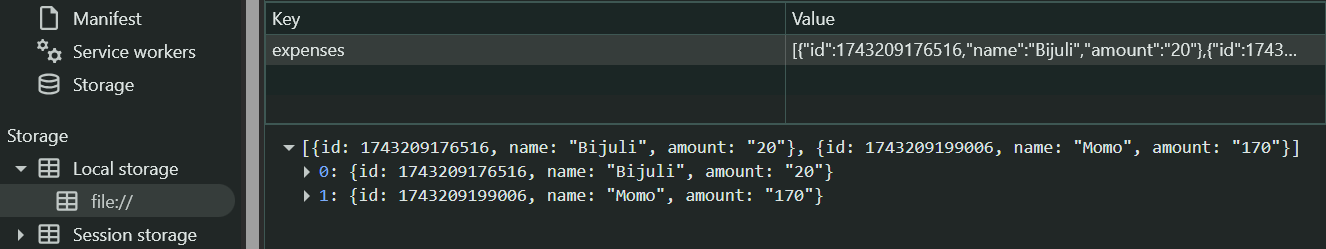
****

1. **Define saveExpenseToLocal function and call in side the submit event**

*(save the array in local storage, before it --- do* ***JSON.stringify(array)*** *)*

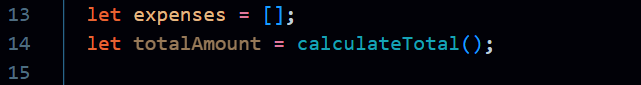
****

****

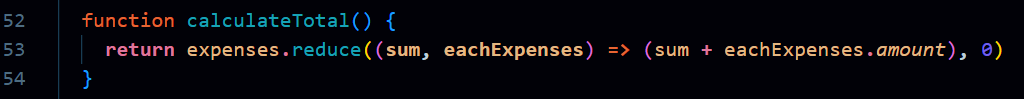
****

1. **Work on Total Expenses [ Calculate Total 🡺 Display Total ]**

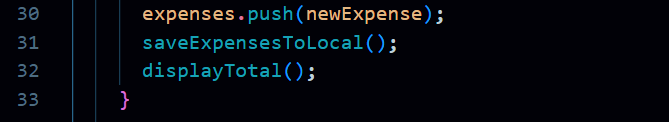
1. **Define calculateTotal function with return and store it in totalAmount variable**

****

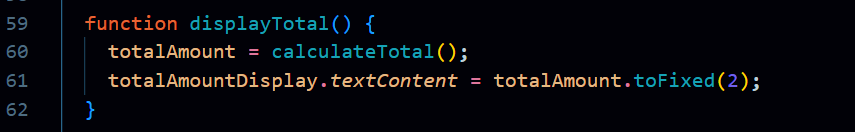
*(do Calculate total amount using* ***array.reduce()*** *method)*

****

1. **Define displayTotal() and call it inside the form-submit event**

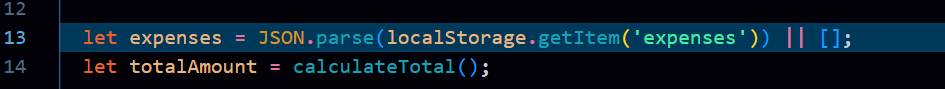
****

*(bring the calculateTotal first then show it in HTML)*

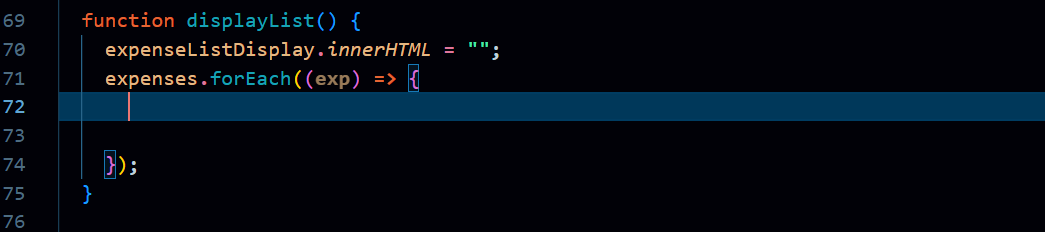
****

1. **To save data permanently in local storage [do getItems & re-store in the save array]**

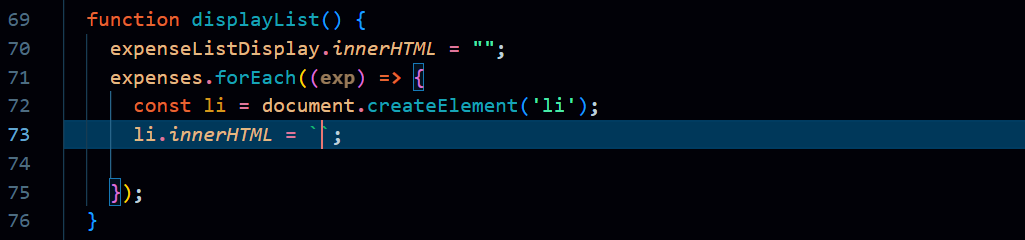
*(re-convert the string data from local-storage in to JS object to re-assign in the same array)*

****

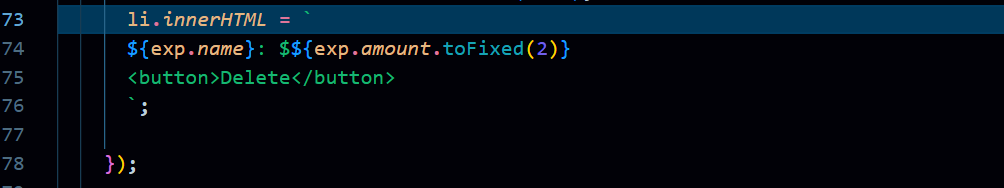
1. **Display all data *[ Define displayList() & apply arr.forEach() for each item of expense array]***

****

1. **Declare li  *then* add inner HTML in *li***

****

1. **Pass back Quots for** *li.innerHTML* **with*🡺 expense name, amount & delete button*** *inside*

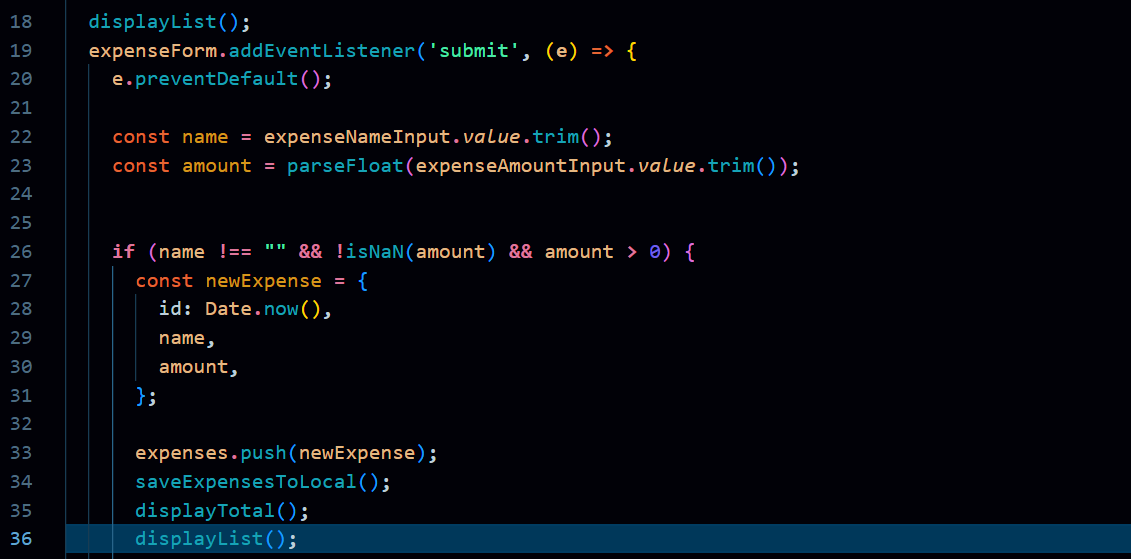
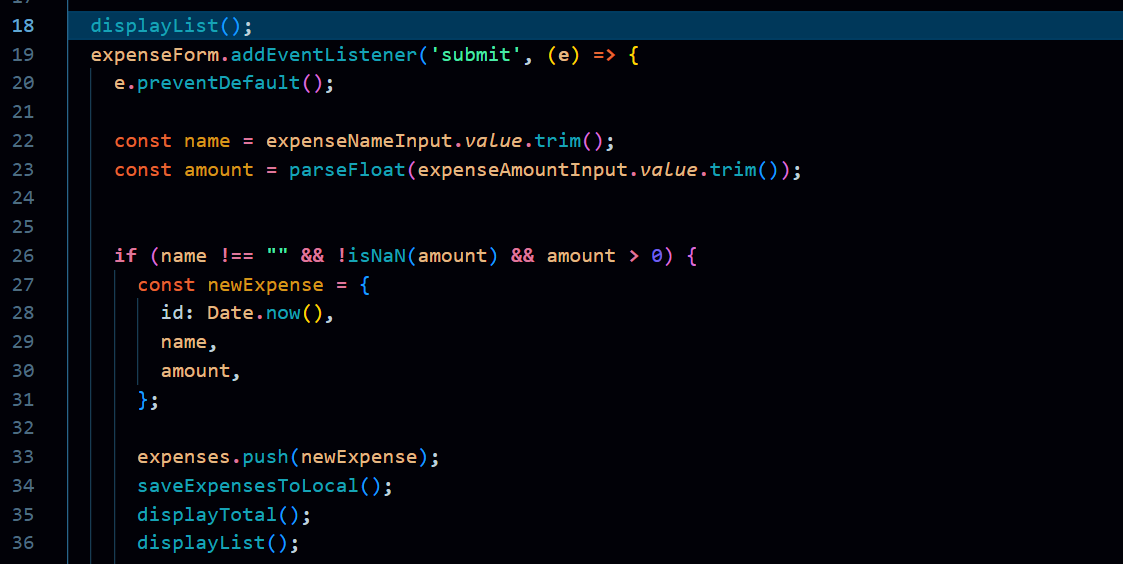
****

1. **Append the li in HTML to display [ *expenseListDisplay.appendChild(li) ] inside forEach()***

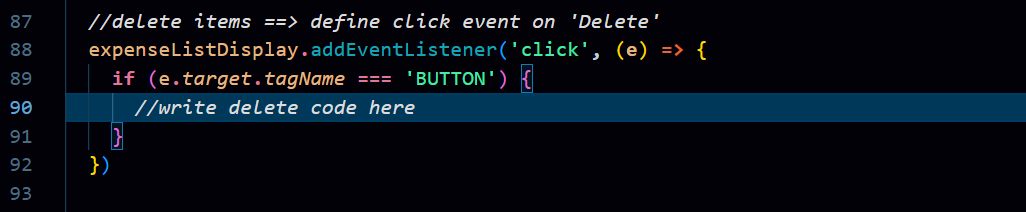
****

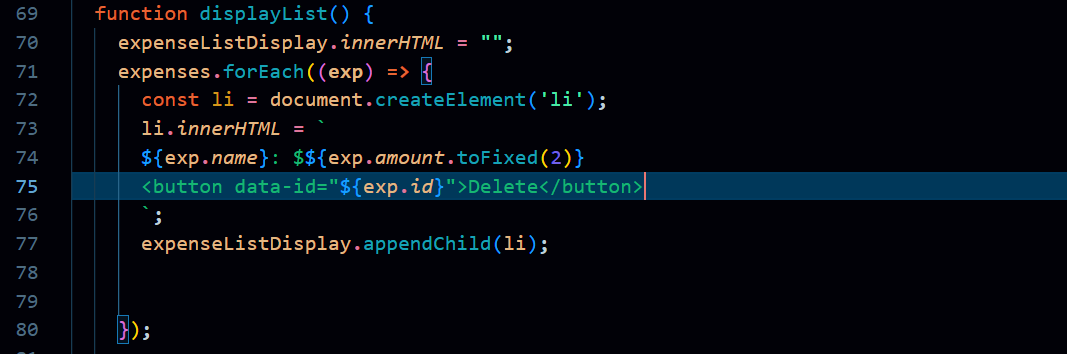
1. **To display all data constantly after-submit and while-pageload *[ call displayList() ]***

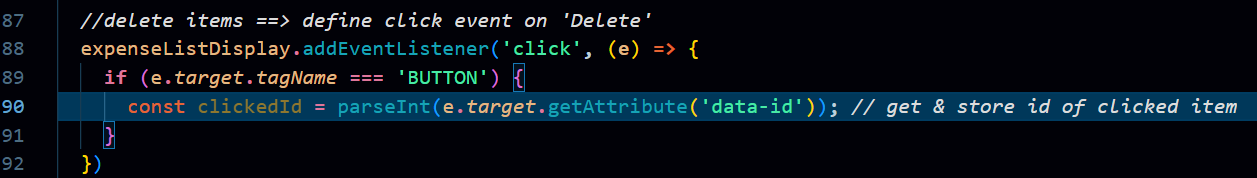
*(define* ***displayList()*** *function and call it* ***inside and outside the submit event****)*

******

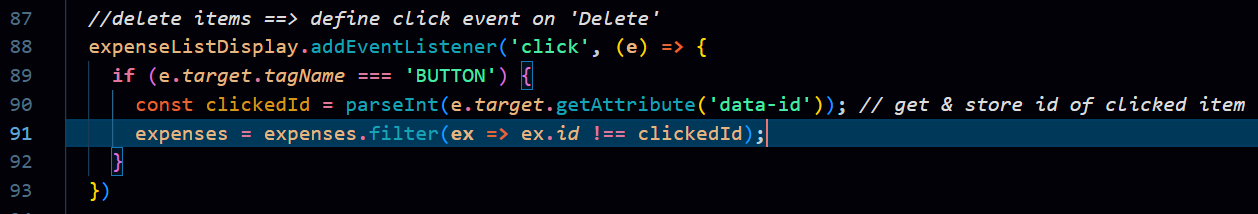
1. **Delete Items [*add click event on delete button with if-clause*]**

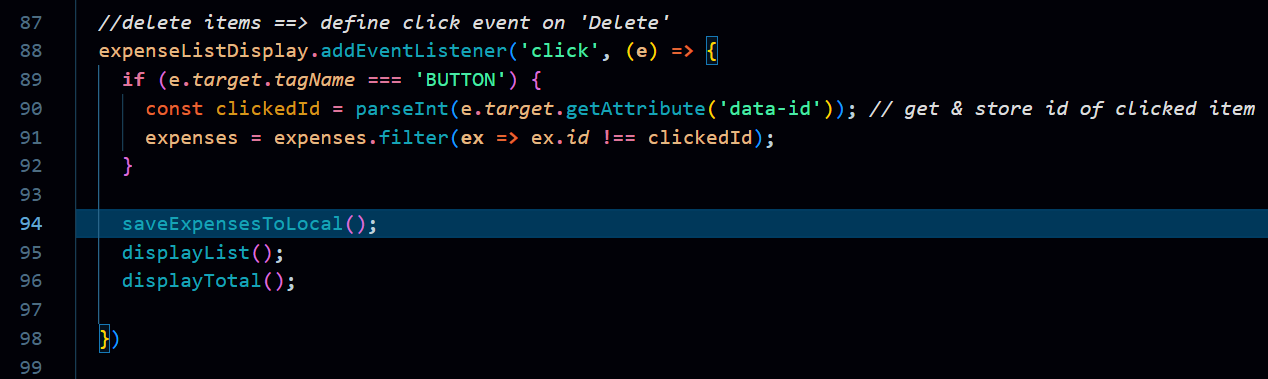
****

1. **Add data-id attribute on delete button in displayList() function **
2. **Declare variable of id by [getting attribute of delete button] for clicked expenses 🡺inside the expenseListDisplay-clicked-event , #*note: convert id into integer***

****

1. **Apply arr.filter() 🡺 filter in the same array 🡺 save only which [clicked-id !== expense-id]**

****

1. **Then after click on Delete *(filter happens)* saveToLocalStorage, displayList & displayTotal**