Classes:

In this part you will create 2 classes.

1. Create a parent class and name it Employee
2. Add a static variable name it number of Employees which you have to increase it when you add a new employee
3. Add a static method to increase the number of employees.
4. An employee has a name, email, age, address, years of experience and salary.
5. Create a constructor and set salary to 1000 (get the other attributes values from the user)
6. add a print method to print the employee information.
7. Add a raise salary methos to increase the employee salary by 1.5.
8. Create a new class Manager that inherits from Employee class.
9. Add a static variable name it number of managers which you have to increase it when you add a new manager
10. Add a static method to increase the number of managers.
11. Manager has a name, email, age, address, years of experience, salary, department.
12. Create a constructor an set salary to 1500.
13. Override the method print.
14. And override the method raise salary to raise the salary by 2.
15. Create a company array and create 2 managers and 3 employees,don’t forget to increase number of employee and managers in both classes.
16. Use Instance of to check if the object is manager and call raise salary.

Objects:

1. Create an object of a user.
2. A user has name, email and password.
3. Add getters and setters to name.
4. For the password property set writable, enumerable, and configurable to false.
5. Add nationality property and set value to Lebanese.
6. Add a function to print the user information (name, email, language)
7. Use a function that returns the object keys.
8. Delete nationality from user object.
9. Now create a prototype of user object.

DOM manipulation:

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Shopping list example</title>

    <style>

        li {

            margin-bottom: 10px;

        }

        li button {

            font-size: 8px;

            margin-left: 20px;

            color: #666;

        }

    </style>

</head>

<body>

    <h1>My shopping list</h1>

    <div>

        <label for="item">Enter a new item:</label>

        <input type="text" name="item" id="item">

        <button>Add item</button>

    </div>

    <ul>

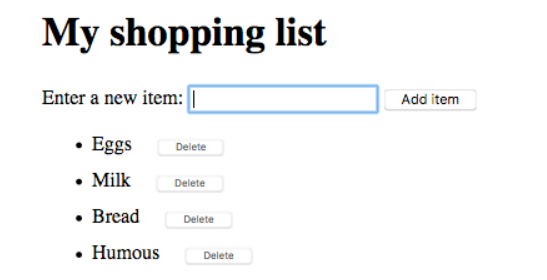
    </ul>

    <script>

    </script>

</body>

</html>



1. Create three variables that hold references to the list ([<ul>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul)), [<input>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input), and [<button>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/button) elements.
2. Create a [function](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Functions) that will run in response to the button being clicked.
3. Inside the function body, start off by storing the current [value](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement#properties) of the input element in a variable.
4. Next, empty the input element by setting its value to an empty string — ''.
5. Create three new elements — a list item ([<li>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li)), [<span>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span), and [<button>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/button), and store them in variables.
6. Append the span and the button as children of the list item.
7. Set the text content of the span to the input element value you saved earlier, and the text content of the button to 'Delete'.
8. Append the list item as a child of the list.
9. Attach an event handler to the delete button, so that when clicked it will delete the entire list item it is inside.