TECH ODYSSEY

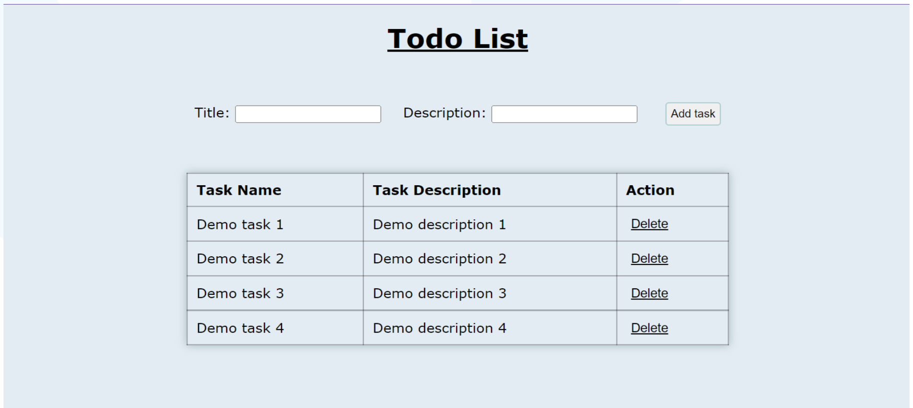
WEB TEAM - WEEK 3

Hello peeps! Hope you had fun with the second week's task where you added styles to the basic to-do list HTML page. We will be building on that this week. For your week 3 task, you will be learning Javascript and how to include it with your existing web page. Let's get started!

Task 3

The aim is to make to-do list functional.When the user clicks the “Add task” button, the title and description of the task must be added in the table and when the user clicks the “Delete” button , the task must be removed from the table.

(**Hint**: var , getElementById , functions, Events - onclick , innerHTML , parentNode - removeChild)



Todo List

Title: [ ] Description: [ ] Add task

Task Name	Task Description	Action
Demo task 1	Demo description 1	Delete
Demo task 2	Demo description 2	Delete
Demo task 3	Demo description 3	Delete
Demo task 4	Demo description 4	Delete

Keep learning, and Happy coding ;D

JAVASCRIPT:

1. What is Javascript?

JavaScript is a programming language that powers the dynamic behavior on most websites. Alongside HTML and CSS, it is a core technology that makes the web run.

Websites for reference:
https://www.w3schools.com/js/default.asp

2. Why do we need Javascript and how to add?

JavaScript is needed to program the behavior of web pages
In HTML, JavaScript code is inserted between <script> and </script> tags.Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

```
<script src="myscripts.js"></script> // points to external js file
```

Reference : *https://www.w3schools.com/js/js_whereto.asp*

3. console.log()

The console.log() method is used to log or print messages to the console. It can also be used to print objects and other info.

```
console.log('Hi there!');
```

4. Variables

A variable is a container for data that is stored in computer memory. It is referenced by a descriptive name that a programmer can call to assign a specific value and retrieve it.

```
let name = "Tammy";
const found = false;
var age = 3;
console.log(name, found, age);
// Prints: Tammy false 3
```

- **var** is used in pre-ES6 versions of JavaScript.
- **let** is the preferred way to declare a variable when it can be reassigned.
- **const** is the preferred way to declare a variable with a constant value.

Reference : *https://www.w3schools.com/js/js_variables.asp*

5. What are Javascript data types?

1) Numbers
2) Strings
3) Boolean
4) Null
5) Undefined
6) Object
7) BigInt
8) Symbol

Reference : https://www.w3schools.com/js/js_datatypes.asp

6. typeof Operator

to find the data type of a JavaScript variable

```
typeof  "John"          // Returns "string"
```

7. Arrays and Objects

Arrays are lists of ordered, stored data. They can hold items that are of any data type. Arrays are created by using square brackets, with individual elements separated by commas.

```
let numberArray = [0, 1, 2, 3];
```

An **object** is a built-in data type for storing key-value pairs. Data inside objects are unordered, and the values can be of any type.
The reserved keyword this refers to a method’s calling object, and it can be used to access properties belonging to that object.

```
const student = {
  name: 'Sheldon',
  score: 100,
  grade: 'A',
}

console.log(student)
// { name: 'Sheldon', score: 100, grade: 'A' }
```

JavaScript objects are mutable, meaning their contents can be changed, even when they are declared as const. New properties can be added, and existing property values can be changed or deleted.

Reference : https://www.w3schools.com/js/js_objects.asp
Reference : *https://www.w3schools.com/js/js_arrays.asp*

8. Methods

Methods return information about an object, and are called by appending an instance with a period ., the method name, and parentheses.

Reference : https://www.w3schools.com/js/js_number_methods.asp
Reference : https://www.w3schools.com/js/js_date_methods.asp

9. Conditional Statements

Conditional statements are used to decide the execution flow, based on different conditions.
When the condition is true, specific action is performed, and if the condition is false, another action will perform.

```
if (condition1) {
  // block of code to be executed if condition1 is true
} else if (condition2) {
  // block of code to be executed if the condition1 is false and
condition2 is true
} else {
  // block of code to be executed if the condition1 is false and
condition2 is false
}
```

Reference : http://bit.ly/3EwTYQL

10. Switch case :

The switch statement is used to perform different actions based on different conditions.

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

Reference : *https://www.w3schools.com/js/js_switch.asp*

11. Loops:

Types of Loops in Javascript

- **for**
- **for...in**
- **while**
- **do while**

Reference : *http://bit.ly/3OuN5DY*

12. Functions

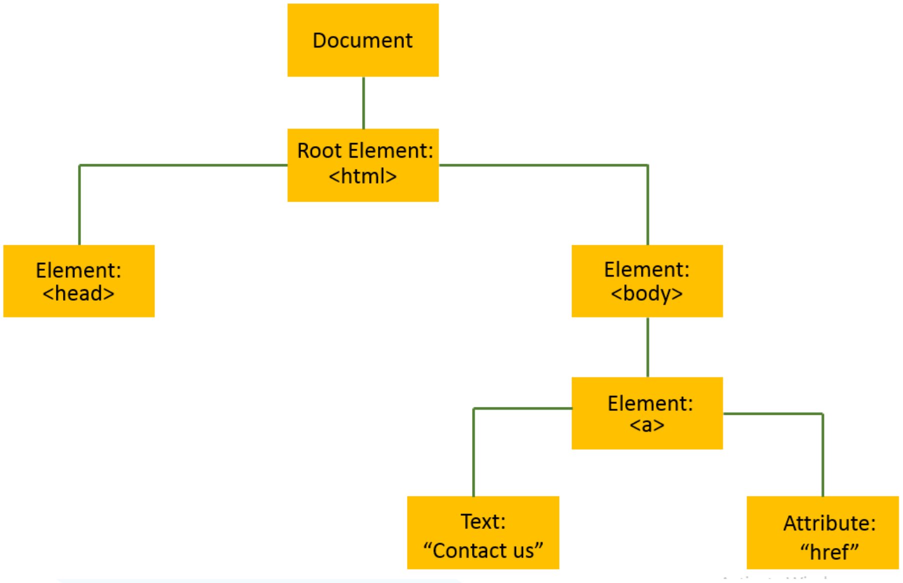
Function is a group of reusable code which can be called anywhere in the program.This eliminates the need to rewrite the same code. Functions allow a programmer to divide a big program into a smaller and manageable function.

```
// Named function
function rocketToMars() {
  return 'BOOM!';
}


// Anonymous function
const rocketToMars = function() {
  return 'BOOM!';
}


// Arrow function with two parameters
const sum = (firstParam, secondParam) => {
  return firstParam + secondParam;
};

console.log(sum(2,5)); // Prints: 7
```

Reference : [*https://www.w3schools.com/js/js_functions.asp*](https://www.w3schools.com/js/js_functions.asp)

13. Javascript DOM

Abbreviated as Document Object Model, it is used to manipulate content, style and structure of the targeted elements. It is one of the most unique and useful tools of JavaScript.When a web page is loaded, the browser creates a Document Object Model of the page.
The HTML DOM model is constructed as a tree of Objects
It is the root element that represents the HTML document.



Source : http://bit.ly/3EU7Dmq
Reference : https://www.w3schools.com/js/js_htmldom.asp

14. The Document Object

When an HTML document is loaded into a web browser, it becomes a document object.

- The document object is the root node of the HTML document.
- The document object is a property of the window object.
- The document object is accessed with:
- window.document or just document

Ref : https://www.w3schools.com/jsref/dom_obj_document.asp

15. Styling an element and Manipulating element

To manipulate HTML elements and its style

- Finding HTML elements by id
- Finding HTML elements by tag name
- Finding HTML elements by class name
- Finding HTML elements by CSS selectors
- Finding HTML elements by HTML object collections

```
const target = document.getElementById( "id")
const targetList = document. getElementsByClass("class")
const targetTag = document. getElementsByTagName("li")
const targetByQueryID = document.querySelector("#id")
const targetByQueryTag = document.querySelector("div")
const  targetQueryAll = document.querySelectorAll(".class")
```

```
target.style.color = “red”
```

Reference : https://www.w3schools.com/js/js_htmldom_css.asp

16. Playing with Elements

Adding and Removing Nodes (HTML Elements)

```
const parent = document .getElementById( "bysomeid" )
const child = document.createElement( 'div' )
child.setAttribute( 'id', "somenewid" )
console.log(child.getAttribute('id’))
child.removeAttribute('id')
parent.append (child)
child.remove( )
```

Ref : *https://www.w3schools.com/js/js_htmldom_nodes.asp*
Ref : *https://www.w3schools.com/js/js_htmldom_elements.asp*
Ref : *http://bit.ly/3U2HDto*

17. Events

Events are signals fired inside the browser window that notify of changes in the browser or operating system environment. Programmers can create event handler code that will run when an event fires, allowing web pages to respond appropriately to change.

Keyboard events

- **keydown** events are fired when the key is first pressed.
- **keyup** events are fired when the key is released.
- **keypress** events are fired when the user presses a key that produces a character value

Mouse events

- **click** events are fired when the user presses and releases a mouse button on an element.
- **mouseout** events are fired when the mouse leaves an element.
- **mouseover** events are fired when the mouse enters an element's content.
- **mousedown** events are fired when the user presses a mouse button.
- **mouseup** events are fired when the user releases the mouse button.

Reference : *http://bit.ly/3i6eSyy*
Reference : *https://www.w3schools.com/jsref/dom_obj_event.asp*

18. Event Listeners

- **.addEventListener()**

  We can tell our code to listen for an event to fire using the **.addEventListener()** method. The advantage of this is that you can add many events to the event target without overwriting existing events. Two arguments are passed to this method: an event name as a string, and the event handler function.

  ```
  document.getElementById("myBtn").addEventListener("click",
  displayDate);
  ```

- **.removeEventListener()**
  To tell the code to stop listening for that event to fire, we can use the **.removeEventListener()** method. This method takes the same two arguments that were passed to .addEventListener(), the event name as a string and the event handler function.

  ```
  element.removeEventListener("mousemove", myFunction);
  ```

  Ref: https://www.w3schools.com/js/js_htmldom_eventlistener.asp

References to Learn Javascript :

- https://www.youtube.com/watch?v=PkZNo7MFNFg
- https://www.w3schools.com/js/default.asp
- https://developer.mozilla.org/en-US/docs/Web/JavaScript

THANK YOU!