JavaScript Roadmap (13 Weeks)

Week 1: JavaScript Fundamentals

Day 1:

Lesson 1: Types of Variables and Data Types

In this introductory lesson, we will explore the fundamental concepts of JavaScript variables and data types.

1. Variables: Understanding the Types and Differences
    - var
    - let
    - const

2. Output Methods:
    - console.log()
    - document.write()
    - alert()

3. Data Types of Variables: Focusing on Primitive Data Types
    - Number
    - String
    - Boolean
    - Undefined
    - Null

4. Non-Primitive Data Types
    - Array
    - Object

Day 2:

Lesson 2: Operators and Conditional Statements

Day 2 delves into the realm of JavaScript operators and conditional statements.

1. Operators:
    - Relational Operators (=, ==, ===, >, >=, <, <=)
    - Logical Operators (&&, ||,!, !=, !==)
    - Arithmetic Operators (+, -, *, /, %)
    - Unary Operators:
    - Increment Operator (++)
    - Decrement Operator (--)

2. Conditional Statements:
    - If Statement
    - If-else Statement
    - if-else-if Statement

3. Getting User Input:
    - Using prompt()
    - Implementing Conditional Questions

Day 3:

Lesson 3: switch() case and Practice Questions

Day 3 introduces the switch() statement and provides practice questions to reinforce your understanding.

1. switch() case:
    - Leveraging switch for conditional execution
    - Practice Questions

Day 4-5:

Lesson 4: Loops - While, Do-While, For

Day 4 and 5 focus on control flow with different loop structures.

1. Loop Types:
    - while-loop
    - do-while-loop
    - for Loop

2. Nested Loops: Understanding multiple loop structures
3. Loop Questions: Practical exercises to solidify your understanding

By the end of Week 1, you will have a solid foundation in JavaScript's core concepts, including variables, data types, operators, conditional statements, and loops. These are essential building blocks for your journey into JavaScript programming

Week 2: JavaScript Web Development and DOM Manipulation

Day 6:

Lesson 5: Creating a Website with JavaScript

On the first day of Week 2, students will embark on a practical project to create a website using JavaScript. We'll explore various methods and techniques to manipulate the Document Object Model (DOM) and enhance web content.

1st Project: Creating a Website with `createElement`

Website Link for Reference:

https://www.w3schools.com/w3css/tryw3css_templates_gourmet_catering.htm

Techniques Covered:

- innerHTML
- innerText
- textContent
- setAttribute
- getAttribute
- createAttribute
- appendChild
- append

Day 7:

Lesson 6: Inline Mouse Events and Function Declaration

Day 7 focuses on introducing students to inline mouse events and function declaration, key elements in JavaScript programming.

- **Inline Events:** Understanding event handling in HTML
- **Function Declaration:** Introducing functions, with a specific focus on creating a "Counter" function.

Day 8:

Lesson 7: Non-Primitive Data Types and Games

Day 8 begins with an exploration of non-primitive data types. Afterward, we delve into interactive games that will reinforce your understanding of if-else and if-else-if statements, callback functions, and generating random numbers.

1. Non-Primitive Data Types: In-depth examination of non-primitive data structures.
2. Generating Random Numbers from an Array:
    - Utilizing **`Math.random()`**
    - Rounding with **`Math.floor()`**, **`Math.round()`,** and **`Math.ceil()`**

3. Interactive Games:
    - **"Head Tail"** game development, incorporating if-else statements and inline events.

Day 9:

1st Task: Rock Paper Scissors

On Day 9, students will engage in a hands-on task to implement the popular "Rock Paper Scissors" game, applying the concepts learned thus far.

Day 10:

Lesson 7: Advanced JavaScript Techniques and DOM Manipulation

The final day of Week 2 explores advanced JavaScript concepts and the Document Object Model (DOM).

- **Array Methods:** Introduction to essential JavaScript array methods.
- **DOM (Document Object Model):** Understanding the structure and manipulation of web documents.
- **`addEventListener`:** Implementing event listeners for interactive web elements.
- **`classList.add`, `classList.remove`, `classList.contains`:** Managing CSS classes dynamically.
- **Spread Cards:** Using loops to add and remove classes for card manipulation.
- **Sidebar Opening and Closing:** Implementing event listeners to control sidebar functionality.

Week 2 equips students with the practical skills necessary for web development and enhances their understanding of JavaScript's role in creating dynamic and interactive web applications.

Week 3: Building Advanced Web Projects with JavaScript

Day 11-13:

In the first half of Week 3, students will embark on a comprehensive project to build a fully functional website with JavaScript.

2nd Project: Alma Website

Project Link for Reference: https://rb.gy/97nvn

Techniques Covered:

- Scroll Event

Over the course of three days, students will work on creating the "Alma Website." This project will involve implementing various logics and functionalities to enhance the website's interactivity and user experience.

Day 14-15:

The latter half of Week 3 introduces a new project that will challenge students to build a dynamic dashboard using JavaScript. This project is designed to further strengthen their understanding of web development and JavaScript application.

3rd Project: Dashboard

Project Link for Reference: https://demo.dashboardpack.com/architectui-html-pro/

Techniques Covered:

- Sidebar
- Accordion
- Dropdowns
- Notifications
- Carousal Slider
- Events
- Loops

Week 3 marks a significant milestone in the JavaScript roadmap, as students apply their knowledge to real-world web development projects, solidifying their skills and gaining hands-on experience in creating interactive and dynamic websites and dashboards.

Week 4: Advanced Dashboard Development and Class Manipulation

Day 16-18:

3rd Project: Dashboard (Continuation)

Project Link for Reference: https://demo.dashboardpack.com/architectui-html-pro/

During the first three days of Week 4, students will extend and enhance the functionality of the dashboard, building on the work started in the previous week.

Day 19-20:

In the latter part of Week 4, students will explore new concepts and undertake a fresh project that focuses on class manipulation and sliders.

Lesson 8: Class Manipulation and For-Of Loop

- **Class Add and Remove:** Understanding the manipulation of CSS classes using JavaScript.
- **For-Of Loop:** Leveraging a loop to interact with and manipulate elements in the DOM.

4th Project: Building Slider Components

This project introduces students to the creation of interactive slider components with various features.

Techniques Covered:

- **Opacity Slider:** Creating a slider to control the opacity of elements.
- **Moving Slider:** Implementing a slider for controlling the position of elements.
- **Background-Image Slider:** Developing a slider that changes background images dynamically.

Week 4 offers students an opportunity to continue their advanced dashboard project, further honing their skills in web development, and also introduces new concepts related to class manipulation and slider components. This week's lessons and projects provide valuable experience in creating dynamic and interactive web content.

Week 5: Advanced JavaScript Concepts and Project Development

Day 21:

2nd Task: Handling Pop-Up and Input Value

On the first day of Week 5, students will delve into the practical aspects of working with pop-up dialogs and handling input values.

- **Pop-Up Task:** Understanding how to retrieve values from input elements using JavaScript.
- **Input Value**: Implementing an interactive feature that utilizes input values to control image blur and un-blur effects.

5th Project: Meme Maker

This project enables students to create custom internet memes, combining images and text through JavaScript, adding a creative and humorous dimension to their coding skills.

Techniques Covered:

- **Image Array:** Build an array of image URLs for random selection.
- **DOM Manipulation:** Update the DOM with selected images.
- **Error Handling:** Ensure a smooth user experience with error handling.
- **User Interaction:** Allow users to select images, add text, and preview memes.

Day 22:

Lesson 9: Timer Functions

Day 22 introduces students to built-in timer functions in JavaScript, along with practical examples.

- **`setInterval()`:** Learning how to set up recurring timed actions.
- **`setTimeout()`:** Implementing one-time timed events.
- **`clearInterval()`:** How to clear intervals and timeouts.

6th Project: Loader

Techniques Covered:

- **`setInterval`** for creating timed loading animations.
- Controlling animations with **`clearInterval`.**
- Implementing conditional logic with **`if-else`** statements.
- Manipulating the **DOM** dynamically.
- Enhancing the **user experience** with visual feedback during loading.
- **Handling errors** for a smoother user experience.

Day 23:

Day 23 focuses on project development, covering projects related to pop-up notifications and dashboard components.

7th Project: Toaster

Techniques Covered:

- `createElement`
- `setTimeOut`
- Get Elements from Array Randomly

Moreover, student will create this project by using techniques which have been used in 1st project named **`createElement`**

8th Project: Dashboard Component – Notifications

Techniques Covered:

- Implement dynamic notifications with **`setTimeout`.**
- Populate notifications with dynamic content from **arrays**.
- Enhance visual appeal by applying **random text colors**.
- **Personalize notifications** by replacing text with input values.

Day 24:

On the 24th day, students will undertake a project centered-around creating a slider carousel.

9th Project: Slider Carousel

Techniques covered:

- JavaScript logic for slide management.
- Auto-play with **`setInterval`.**
- Manual navigation with **`clearInterval`**.
- Navigation controls for user interaction.

Day 25:

The final day of Week 5 introduces a project related to theme customization and dynamic clock displays.

10th Project: Dark and Light Mode Clock Theme

Techniques Covered:

- Fetch current time with the **Date** object.
- Update the digital clock dynamically.
- Calculate and draw analog clock hands.
- Create ticking animations for analog clock hands.
- Continuously update the clock display in real-time.
- Add event listeners for light/dark mode toggling.

Week 5 offers students the opportunity to explore advanced JavaScript concepts such as timers and dynamic content manipulation. They will also be engaged in practical projects that reinforce their skills in web development and interactivity, providing valuable experience in creating various features and components for web applications.

Week 6: JavaScript Project Development and Skill Enhancement

Day 26:

11th Project: Testimonial Boxes

In the first day of Week 6, students will embark on a project aimed at creating testimonial boxes. This project will involve the design and implementation of interactive testimonial displays.

Techniques Covered:

- **Data storage** in objects.
- **Destructuring** for efficient data access.
- Data updates using **`setInterval`.**
- Dynamic HTML element population.
- **Array** iteration for multiple item management.

1st Assignment from Home:

12th Project: Stopwatch

Students will be given their first assignment to work on from home, focusing on developing a stopwatch application.

Techniques Covered:

- Track time, including hours, minutes, seconds, and milliseconds.
- Start, pause, and reset the stopwatch.
- Update the time display dynamically.
- Utilize **`setInterval`** and **`clearInterval`** to control time updates.
- Handle button click events to trigger **stopwatch** actions.

Day 27:

13th Project: Style Change by Range Input

Day 27 introduces a project where students will learn how to change the style of elements using range input. This project focuses on creating interactive and dynamic user interfaces.

Techniques Covered:

- Retrieve photo and range input elements.
- Add event listeners to detect input changes.
- Update styles using the **`style`** property.
- Adjust the **`transform`** property for rotation and scaling.
- Modify the **`filter`** property for visual effects.
- Manipulate **`box-shadow`** and text color properties.
- Continuously update styles based on input changes.
- Implement error handling for style adjustments.

14th Project: Find Bird

In the second project of the day, students will work on an application designed to find and display bird-related information.

Techniques Covered:

- Generate random bird positions.
- Implement timer functionality.
- Add click event listeners to bird images.
- Track and increase the player's score.
- Define game over conditions.
- Allow game resets for multiple rounds.
- Implement error handling for timer and player actions.

Day 28:

15th Project: Catch Insects

Day 28 presents a project where students will create a game called "Catch Insect". This project combines game development with JavaScript skills.

Techniques Covered:

- **Insect Movement:** Create logic to move insects around the screen in random or predefined patterns.
- **Character Control:** Enable player control of the character to catch falling or moving insects.
- **Score and Time Tracking:** Implement a scoring system to keep track of how many insects the player catches.
- **Game over Condition:** Define the end condition, such as reaching a specific score or missing too many insects.
- **Restart Functionality:** Allow players to restart the game for multiple rounds.
- Error Handling: Implement error handling for issues like game over conditions and player actions.

Day 29:

16th Project: Guess the Random Number

On the 29th day, students will undertake a project that involves guessing a random number. This project emphasizes interactive user experiences and logical thinking.

Techniques Covered:

- Implement **event listeners** for user interactions.
- Develop core game logic, including random number generation and tracking guesses.
- Handle errors, including **non-numeric** input.
- Manage game flow, enabling restarts and range adjustments.
- Generate **random numbers** for the game.
- **Display** dynamic messages to inform players.
- Allow game **restarts** for multiple rounds.

Day 30:

17th Project: Guess the Word using Char-code Method

The final day of Week 6 introduces a project where students will work on a word guessing game, incorporating the Char-code method. This project enhances their knowledge of character encoding and user interaction.

Techniques Covered:

- Shuffle words randomly for each game.
- Split words into individual letters, creating a guessing board.
- Allow letter selection by players through button clicks.
- Provide dynamic feedback messages for player guesses.
- Enable game restart for multiple rounds.

Week 6 is filled with a diverse range of projects, from interactive testimonial boxes to games and applications, allowing students to apply their JavaScript skills in practical and creative ways. Additionally, the first assignment from home, the stopwatch, offers a chance for independent project work and skill development.

Week 7: JavaScript Project Development and Advanced Applications

Day 31:

18th Project: Speed Typing Text at Different Levels

In the first day of Week 7, students will engage in a project centered-around speed typing text. This project will incorporate various levels of difficulty to challenge their typing skills.

Techniques Covered:

- **DOM Manipulation:** Access and modify HTML elements, handle events, and apply styles.
- **Game Logic:** Manage word selection, user input matching, scoring, and difficulty levels.
- **Timer & Countdown:** Create a countdown timer displayed to the user.
- **Random Word Selection**: Randomly choose words from a list for typing.
- **Event Listeners:** Utilize event listeners to respond to user input and level selection.
- **Game Flow:** Control the game's progression, handle messages, and update the timer.
- **Difficulty Levels:** Allow users to select different game difficulty levels.
- **Score Tracking:** Keep score and update it when users type words correctly.

2nd Assignment from Home:

19th Project: Typing Speed

Students will receive their second assignment to complete from home, focusing on measuring and improving typing speed.

Techniques Covered:

- **Event Listeners:** Toggle the game's state and user input field based on button clicks.
- **Timing:** Measure user completion time.
- **Word Count:** Calculate the number of typed words.
- **Word Comparison:** Compare typed words with the original message.
- **Random Text Selection:** Display random text prompts for variety.
- **User Interaction:** Start and finish the typing challenge, providing typing speed and accuracy feedback.

Day 32:

20th Project: Mouse Movement and Size Change

Day 32 introduces a project where students will explore mouse movement and its interaction with elements on the web page, including resizing.

Techniques Covered:

- **Cursor Tracking:** The code tracks the position of the mouse cursor using **clientX** and **clientY** properties of the event object, allowing for interaction based on cursor location.
- **Dynamic Element Positioning:** Updates **"move"** element position based on cursor, creating a following effect.
- **Interactive Resizing:** The code allows users to click on specific elements (one elements) to change the size of the element, demonstrating user interaction.

21st Project: Follow Mouse Animation

In the second project of the day, students will work on an animation that follows the mouse cursor, enhancing their understanding of interactive user experiences.

Techniques Covered:

- **Event Handling:** It listens for the **mousemove** event using the **`addEventListener`** method, enabling mouse tracking.
- **Dynamic Element Creation:** New child elements (div elements with the class **"child"**) are generated in response to mouse movement.
- **Element Positioning:** The code calculates and sets the positions of the "container" and **"child"** elements based on the mouse cursor's coordinates.
- **CSS Animation:** CSS keyframes (@keyframes) and animations are defined and applied to the **"container"** and **"child"** elements, creating visual effects.
- **Filter Effects:** Filter effects like blur and hue-rotate are applied to achieve visual transformations.

Day 33:

22nd Project: Double Click Heart (Instagram)

On this day, students will undertake a project to create a double-click heart animation, similar to the one seen on Instagram.

Techniques Covered:

- **Dynamic Element Creation:** Double-clicking creates heart-shaped elements and appends them to element.
- **Element Positioning:** The script places hearts at the double-click coordinates within the element.
- **Counter Update:** It tracks double-clicks (likes) and updates the count in the **"times"** element.

Day 34:

23rd Project: Progress Steps

Day 34 presents a project where students will create progress steps, providing visual indications of progression through a series of steps.

Techniques Covered:

- **Dynamic Highlighting:** The code dynamically highlights completed steps by updating the **"active"** class on circles.
- **Visual Progress Bar:** The progress bar visually reflects the completed steps by adjusting its width.
- **Boundary Control:** It prevents moving beyond the first and last steps, ensuring the active step stays within a valid range.

3rd Assignment from Home:

24th Project: Registration Form

Students will receive their third assignment, this time focusing on the development of a registration form for a web application.

Techniques Covered:

- **Multi-step Form:** The code implements a multi-step registration form with a progress bar and step navigation.
- **Progress Bar:** A dynamic progress bar represents the user's advancement in the form. As the user completes each step, the progress bar updates accordingly.
- **Step Navigation:** Users can navigate between form steps using **"Next"** and **"Previous"** buttons.
- **Validation:** Although not shown in the provided code, this structure is suitable for implementing step-by-step form validation to ensure data accuracy.

Day 35:

25th Project: Star Rating

The final day of Week 7 introduces a project where students will work on creating a star rating system, enhancing their knowledge of user feedback mechanisms.

Techniques Covered:

- Event listeners for **“mouseover”**, **“mouseout”**, and **"click"** to respond to user actions on the stars.
- Interactive star rating.
- Dynamic CSS class changes for star colors.
- User feedback on selected rating.

Week 7 offers students an array of diverse projects, ranging from interactive animations and user interface enhancements to practical applications like typing speed and registration forms. The assignments from home encourage independent project work and the application of learned skills to real-world scenarios.

Week 8: JavaScript Project Development - Quizzes, Drag and Drop, and Interactive Components

Day 36:

26th Project: Quiz App

In the first day of Week 8, students will dive into a project that involves the creation of a quiz application. This project will test their ability to build interactive and educational web content.

Techniques Covered:

- User answer selection with radio buttons.
- Scoring system for **tracking** correct answers.
- Displaying results and an option to restart the quiz.
- Interactive **"Submit"** button for navigation.

Day 37:

27th Project: 50 Quiz App

Day 2 of week 8 introduces an extended project where students will work on a more extensive quiz application, featuring 50 questions. This project allows for further exploration of interactive quizzes and user engagement.

Techniques Covered:

- **Event Handling:** Using event listeners to respond to user interactions like button clicks.
- **Iterative Development:** Incrementally building the quiz functionality by iterating through questions and answers.
- **Conditional Logic:** Checking user answers and determining correctness with conditional statements (if statements).
- **Array Handling:** Managing arrays to store and retrieve quiz data and user answers.
- **User Feedback Logic:** Calculating and displaying the user's score based on correct answers.
- **Dynamic Form Elements:** Managing and handling form elements (radio buttons) for user input.

- **Page Navigation Logic:** Implementing logic for moving between quiz questions using **"Next"** and **"Previous"** buttons.
- **Input Validation:** Ensuring that all questions are answered before allowing submission.

Day 38:

28th Project: Drag and Drop

On the 38th day, students will engage in a project focused on drag and drop functionality. This project will enhance their understanding of user interface interactivity.

Techniques Covered:

- **HTML5 Drag and Drop:** Utilizes the HTML5 Drag and Drop API for drag-and-drop interactions.
- **Event Listeners:** Sets up event listeners for drag-and-drop events like **`dragstart`**, **`dragend`**, **`dragover`**, **`dragenter`**, **`dragleave`**, and **`drop`**.
- **Dynamic CSS:** Adds/removes CSS classes for visual effects during dragging and hovering.
- **Data Transfer:** Uses the **`dataTransfer`** object for data transfer in drag-and-drop.
- **Element Appending:** Appends draggable elements to drop targets upon a valid drop.

Day 39:

29th Project: Draggable Slider

Day 39 presents a project where students will work on creating a draggable slider. This project emphasizes user interaction and dynamic content manipulation.

Students will complete this project as a practice of previous project.

Techniques Covered:

- **HTML5 Drag and Drop:** Utilizes the HTML5 Drag and Drop API for drag-and-drop interactions.
- **Event Listeners:** Sets up event listeners for drag-and-drop events like `dragstart`, `dragend`, `dragover`, `dragenter`, `dragleave`, and drop.
- **Dynamic CSS:** Adds/removes CSS classes for visual effects during dragging and hovering.
- **Data Transfer:** Uses the **`dataTransfer`** object for data transfer in drag-and-drop.
- **Element Appending:** Appends **draggable** elements to drop targets upon a valid drop.

Day 40:

30th Project: Draggable To-Do List

The final day of Week 8 introduces a project where students will create a draggable to-do list. This project combines task management with interactivity, allowing users to organize their tasks dynamically.

Techniques Covered:

- **Drag-and-Drop:** Enables users to drag and drop to-do items between status columns using various drag-and-drop events.
- **Modal for Adding To-Do Items:** Provides a modal for adding new to-do items.
- **Dynamic To-Do Creation:** Dynamically creates new to-do items based on user input.
- **Styling and UI Interactions:** Manages the visual style and user interactions, such as closing to-do items.
- **Event Listeners:** Uses event listeners for drag-and-drop, modal actions, and closing to-do items.

Week 8 offers a diverse range of projects, from quiz applications that test their knowledge of JavaScript to interactive components like drag and drop functionality and draggable sliders. These projects provide valuable experience in building dynamic and engaging web applications.

Week 9: JavaScript Project Development and Advanced Techniques

Day 41:

31st Project: Paragraph Animation

On the first day of Week 9, students will be working on a project focused on paragraph animation. This project explores animating text and content dynamically.

Techniques Covered:

- **Animation:** Achieving a smooth scrolling effect by changing element positions over time.
- **Interval and Timeout:** Using **`setInterval`** for continuous scrolling and **`setTimeout`** for mouse hover events.
- **User Interaction**: Enabling manual start/stop of scrolling and adjusting speed with an input field.
- **Element Positioning:** Calculating and updating the position of scrolling text for a smooth appearance.

4th Assignment from Home:

32nd Project: Animation-On-Scroll

Students will receive their fourth assignment as a project, which involves creating animations triggered by scrolling, a valuable skill for enhancing user experience.

Techniques Covered:

- **Event Handling:** Event listeners respond to the scroll event on the window.
- **DOM Selection and Manipulation:** It selects HTML elements using **`document.querySelectorAll`** and applies the **`classList`** property for DOM manipulation by toggling the **'show'** class.
- **Intersection Detection:** The **`getBoundingClientRect`** method detects if elements are within the viewport, enabling animations when they become visible during scrolling.

Day 42:

33rd Project: Simple Calculator

Day 2 of week 9 introduces a project where students will build a simple calculator, applying their JavaScript skills to create a practical and interactive tool.

Techniques Covered:

- **Mathematical Calculations:** Using **`eval`** to perform calculations.
- **Button Layout Array:** The project employs a two-dimensional array (myKeys) to structure and arrange calculator buttons.
- **String Methods (.substring()):** The .substring() method is used to extract the last character of the current input for evaluation.
- **User Interaction:** Click event listeners are applied to buttons for user input, enabling arithmetic operations and clearing the display.
- **Output Display:** The calculator displays the current input and result in a responsive output area.

34th Project: EventKeyCode

In the second project of the day, students will work with **`EventKeyCode`**, exploring how to handle keyboard events in web applications.

Techniques Covered:

- **Event Handling:** Utilizing event listeners to capture **keydown** events.
- **DOM Manipulation:** Updating the HTML content dynamically in response to events.
- **Template Literals:** Constructing dynamic HTML content using template literals to display the key, keyCode, and code of the pressed key.

Day 43:

Lesson 10: Array Methods - Filter, Map, Find, and Reduce

In Lesson 10, students will learn the following key concepts related to JavaScript array methods:

- **Array Methods Overview:** Understanding the role of array methods in JavaScript and their importance in data manipulation.
- **Filter Method:** Learning how to use the filter method to create a new array with elements that meet specified criteria, effectively filtering data.
- **Map Method:** Exploring the map method and how to apply a function to each element in an array, creating a new array with the results.
- **Find Method:** Understanding how to use the find method to locate the first element in an array that satisfies a given condition, particularly useful for searching.
- **Reduce Method:** Learning about the reduce method and its ability to reduce an array to a single value, often used for calculations and aggregations.

5th Assignment from Home:

35th Project: Suggestion List

Students will receive their fifth assignment as a project, focusing on the creation of a suggestion list, which helps users find relevant information efficiently.

Techniques Covered:

- **Array Filtering:** The **'filter'** method refines suggestions based on user input.
- **Creating and Appending Elements:** New **'li'** elements are created and added to the suggestions list.
- **User Interaction:** Users can click suggestions to fill the search input.

Day 44:

36th Project: Tags Generating

On the 44th day, students will engage in a project aimed at generating tags. This project enhances their ability to create interactive and user-friendly content.

Techniques Covered:

- **User Input:** Users can input tags separated by commas into the text field.
- Splitting Tags: The input is split into an array of tags using the **`split`** method: **`value = value.split(",")`**.
- **Filtering Tags:** The filter method is applied to remove empty or whitespace-only tags: **`value = value.filter(val => val.trim())`.** The **`trim`** method removes leading and trailing whitespace from each tag.
- **Creating Tags:** For each valid tag, a new **`span`** element is dynamically created using the **`createElement`** method, and the tag content is set as its inner HTML.
- **Displaying Tags:** The newly created span elements are added to the **`.tags_main`** container using the **`appendChild`** method.
- **Random Tag Highlighting:** When the "Enter" key is pressed (key code), a function (`**randomColor**`) is called to add a CSS class to a randomly selected tag, highlighting it.

- **Timeout for Highlight Removal:** After a brief timeout using the **`setTimeout`** method, the "active" class is removed, effectively removing the highlight from the tag.

37th Project: Note App

In the second project of the day, students will create a note-taking application, reinforcing their knowledge of dynamic content and user interfaces.

Techniques Covered:

- **Event Handling:** The **"Add"** button click event triggers the **`addNote`** function.
- **Data Management:** Notes are stored and updated in the **Data array**.
- **Array Mapping:** The **map** function generates HTML for each note in the Data array.

Day 45:

38th Project: Tumie Website

In the first project of the day, students will work on creating a website for "Tumie," focusing on e-commerce and shopping cart features.

39th Project: Add to Cart

The second project of the day involves building an "Add to Cart" feature, emphasizing e-commerce functionality and dynamic user interactions.

Techniques Covered:

- **Product Data:**
  - Array 'products' with id, name, and price.
- **Product Rendering:**
  - **`renderProducts`** creates product cards.
  - Displays name, price, "Add to Cart" button.
- **Pagination:**
  - **`renderPagination`** generates page buttons.
  - Click updates page, renders products.
- **Adding to Cart:**
  - **`addToCart`** adds products to cart.
  - Updates quantity and total price.
- **Updating Cart Display:**
  - **`updateCartDisplay`** refreshes cart item display.
  - Shows name, quantity, total price.
- **Event Listeners:**
  - Handle pagination and adding to cart.
- **Comments:**

- Code includes explanations.

Week 9 offers a wide range of projects and assignments, from animations, calculators, and note-taking apps to e-commerce solutions and array methods. This week provides students with valuable experience in enhancing user interfaces and creating interactive web applications.

Week 10: Advanced JavaScript Projects, Multimedia Integration, and Data Security

Day 46 (Continuation from Week 9, Day 45):

38th & 39th Project: Tumie Website - Add to Cart Feature

On the first day of Week 10, students will continue working on the Website “Tumie” project, specifically focusing on adding a shopping cart feature. This project emphasizes e-commerce functionality and user interaction.

Day 47:

40th Project: Sound Generating

In Day 2 of week 10, students will engage in a project involving sound generation. This project explores how to create and control sound through JavaScript, offering an interactive and auditory experience.

Techniques Covered:

- **Audio Playback:** Utilizes the **Audio** element to play sound files.
- **Managing Playback:** Ensures only one sound plays at a time by pausing the previous sound if needed.

Day 48:

41st Project: Audio Music Player

On this day, students will work on creating an audio music player. This project will help them gain expertise in handling audio files and creating an interactive music player interface.

Techniques Covered:

Player Components:

- Displays album art, title, artist, and controls.
- Progress bar for song duration and current time.

JavaScript Handling:

- Manages DOM elements, song details, and controls.
- Handles an array of songs.

Play/Pause:

- Functions for play/pause with icon toggling.

Load Song:

- Updates displayed info, loads audio and art.

Song Navigation:

- Functions for previous and next songs.
- Handles playlist end/beginning.

Event Listeners:

- Listens for play, pause, prev, and next clicks.
- Auto-advances to the next song.

Initial Load:

- Loads the first song on page load.

Responsive Design:

- Adjusts styling for smaller screens.

Comments:

- Code includes comments for clarity.

Day 49:

42nd Project: Password Generator

Day 49 presents a project where students will build a password generator. This project focuses on creating secure and dynamic password generation functionality.

Techniques Covered:

- **Event Handling:** Managing user interactions.
- **Conditional Logic**: Generating passwords based on criteria.
- **Character Generation:** Creating random characters.
- **DOM Manipulation:** Updating the HTML.
- **Error Handling:** Validating input and displaying messages.
- **Clipboard Copy:** Copying passwords to the clipboard.

6th Assignment from Home:

43rd Project: Form Validation

Students will receive their sixth assignment as a project, centered-around form validation. Form validation is a critical skill for ensuring data integrity in web applications.

Techniques Covered:

- **Form Validation:** Implementing client-side form validation.
- **Event Handling:** Handling form submission and input events.
- **Conditional Statements:** Using conditional statements **(if)** for validation checks.
- **DOM Manipulation:** Accessing and updating the DOM to display error messages.
- **Input Validation:** Checking for empty fields, valid email format, and a valid phone number.
- **Select Dropdown Validation:** Ensuring a valid option is selected from a dropdown menu.

Day 50:

44th Project: Drink Water

The final day of Week 10 introduces a project where students will create a "Drink Water" application, emphasizing user engagement and tracking of daily water intake.

Techniques Covered:

- Selects DOM elements with **`querySelector`** and **`querySelectorAll`**.
- Handles clicks on small cups, updating their status.
- Dynamically updates the big cup and calculates remaining goal/percentage.
- Allows users to click on cups, updating appearance and progress bar.
- Ensures proper handling for already filled cups.
- Sets a 2-liter water goal.
- Displays remaining liters and updates goal percentage.
- Manages visibility of percentage and remaining liters.
- Hides info when the goal is achieved.

Week 10 offers students a variety of advanced projects, including e-commerce features, sound generation, audio players, password generators, and form validation. These projects help students gain hands-on experience in creating interactive and functional web applications.

Week 11: Advanced JavaScript Applications and Data Management

Day 51:

45th Project: Bookmark Application

The first day of Week 11 kicks off with a project focused on building a bookmark application. This project emphasizes the organization and management of web bookmarks.

Techniques Covered:

- **Form Handling:** Prevent default form submission behavior using **`e.preventDefault()`**.
- **Conditional Logic:** Differentiate between adding new bookmarks and editing existing ones.
- **Creating Elements:** Dynamically create HTML elements (li, a, div) for bookmarks.
- **Editing Elements:** Enable users to edit bookmark entries and update the DOM.
- **Removing Elements:** Delete bookmarks from the list by removing the corresponding item from the DOM.
- **Data Validation:** Implement basic input validation to check for empty fields.

Day 52:

46th Project: My Book App

On this day, students will engage in a project to create a book list application. This project enhances their skills in data management and interactive content development.

Techniques Covered:

- **Selectors:** Utilize **`document.querySelector`** and **`document.querySelectorAll`** for advanced CSS selector-based element selection.
- **Element Creation:** Dynamically generate HTML elements using document.createElement.
- Class Manipulation: Manipulate classes through element. **`classList`** for adding, removing, or toggling.
- **Appending Elements:** Append child elements in the DOM using **`element.appendChild`**.
- **Event Handling:** Listen for specific events on elements with **`addEventListener`**.
- **Node Insertion:** Use **`insertBefore`** to insert nodes before a specified reference node
- **Input Validation:** Ensure user input is provided before creating dynamic elements.
- **Alerts:** Provide user feedback or information through alerts using alert.

Day 53:

47th Project: Catch Element

On this day, students will work on a project involving element selection and manipulation. This project explores advanced DOM (Document Object Model) interaction.

Techniques Covered:

- **Randomization:** Generate random positions for elements.
- **Scoring System:** Implement a scoring mechanism.
- **Asynchronous Actions:** Use timeouts for timed actions.
- **Position Tracking:** Display and update element coordinates.
- **Interval-Based Animation:** Create element animations using **`setInterval`**.

// Advanced JavaScript //

Day 54:

Lesson 11: Local Storage

In Lesson 11, students will receive instruction on Local Storage, a crucial concept for data management in web applications.

Topics Covered:

- Introduction to Local Storage
- How to store and retrieve data using Local Storage
- Examples demonstrating the practical application of Local Storage

48th Project: Student Data

A project where students will use Local-Storage to manage and retrieve student data efficiently.

Techniques Covered:

- **Local storage** usage for data persistence.
- Form validation to ensure complete data entry.
- Dynamic element creation based on user input.
- Managing data using JavaScript objects.
- **Array iteration** for data manipulation.
- Conditional statements for data validation.
- Implementing **data removal** and updating local storage

Day 55:

49th Project: To-do List

The final day of Week 11 presents a project where students will build a To-do List application that utilizes Local-Storage for persistent data storage.

Techniques Covered:

- **Add Todo Function:**
  - Creates new todos with completion and deletion buttons.
  - Adds todos to the DOM and saves them locally.
- **Delete Todo Function:**
  - Handles todo deletion with event delegation.
  - Identifies the clicked button (complete or trash).
- **Filter Todo Function:**
  - Filters todos based on selected options (all, completed, uncompleted).
- **Local Storage Functions:**
  - `**saveLocalTodos**`: Saves todos to local storage.
  - `**removeLocalTodos**`: Removes todos from local storage.
- **Get Todos Function:**
  - Retrieves and displays todos from local storage on page load.

Week 11 focuses on advanced JavaScript applications, data management, and data storage techniques, with a specific emphasis on Local Storage. Students will gain hands-on experience in creating interactive web applications and efficiently managing data through Local Storage.

Week 12: Advanced JavaScript Projects and API Data Fetching

Day 56-57:

50th Project: Seats Booking

In the first two days of Week 12, students will embark on a project to create a seats booking application. This project focuses on user interaction and reservations management.

Techniques Covered:

- **Set Movie Data:**
  - Stores selected movie index and price in local storage.
- **Update Selected Count:**
  - Calculates and updates the count and total price based on selected seats.
  - Stores selected seat indices in local storage.
- **Movie Select Event:**

    - Listens for changes in the movie selection dropdown.
    - Updates ticket price and calls **`setMovieData`** and **`updateSelectedCount`**.
- **Seat Click Event:**
    - Listens for clicks on seats that are not occupied.
    - Toggles the "selected" class on the clicked seat and updates the count and total.
- **Event Delegation:**
    - Utilizes event delegation on the container to efficiently handle seat click events.
- **Initial Count and Total:**
    - Sets the initial count and total based on the stored data or defaults.

Day 58:

51st Project: Login and Sign-Up Form

On the third day, students will work on a project to create a login and sign-up form. This project emphasizes user authentication and registration processes.

Day 59:

Lesson 12: API Data Fetching in JavaScript

Day 4 introduces Lesson 12, where students will gain a comprehensive understanding of API data fetching in JavaScript.

Techniques Covered:

- **API Communication:**
    - Bridge between software systems for information sharing.
    - Operates on a request-response mechanism.
- **API Characteristics:**
    - Standardized protocols for structured communication.
    - Facilitates seamless data exchange between applications.
    - Provides functionality access without internal details understanding.
- **API Purpose:**
    - Enhances interoperability between different systems.
    - Promotes modularity and code reusability.
    - Enables third-party integration and scalability.
    - Ensures secure exposure of functionalities.
    - Boosts developer productivity with well-defined interfaces.
    - Contributes to technology ecosystem growth.

52nd Project: Jokes API

Students will work on a project that involves fetching data from a Jokes API, putting into practice what they've learned about data retrieval.

Techniques Covered:

- **Event Listener:**
  - Listens for a click on the **"Get Another Joke"** button **(jokeBtn)** to trigger the **`generateJoke()`** function.
- **Initial Joke Generation:**
  - Calls **`generateJoke()`** on page load to show an initial joke.
- **`generateJoke` Function (async/await):**
  - Asynchronous function fetching jokes from 'https://icanhazdadjoke.com'.
  - Uses await to handle the API request and response.
  - Updates the joke container **(jokeEl)** with the retrieved joke.
- **Alternative `generateJoke` Function (.then()):**
  - An alternative **`generateJoke`** function using the **.then()** syntax will be provided.

Day 60:

53rd Project: Weather API

The final day of Week 12 introduces a project where students will work on an application to fetch weather data from an API, enhancing their knowledge of data acquisition.

Techniques Covered:

- **Initial Weather Display:**
  - Fetch Lahore weather data from **OpenWeatherMap** API.
  - Extract temperature, icon, description, and wind speed.
  - Update HTML elements with fetched data.
- **Event Listener Setup:**
  - Add click event listener to search button (btn).
  - Check if input field is not empty on button click.
  - Fetch weather data for entered city from API.
  - Update HTML elements with newly fetched data.
- **API Data Processing:**
  - Process API response for temperature (in Celsius), icon, description, and wind speed.
  - Dynamic HTML Update:
  - Update inner HTML of elements (city_name, temp, icon, description, speed) with fetched data.
- **Icon Image Display:**
  - Dynamically generate URL for weather icon image.
  - Set it as the source for the icon element.
- **Error Handling:**
  - Display alert if input field is empty, prompting user to enter a city name.
- **Temperature Conversion:**
  - Converts temperature from Kelvin to Celsius for display.
- **API Key:**
  - Uses the **OpenWeatherMap** API key (f65e32faa80c40a876bd4112cd36e525) for authentication.

7th Assignment from Home:

54th Project: Movie App API

Students will receive their first assignment as a project to complete from home, focusing on the development of a movie app that fetches data from an API source. This assignment reinforces the practical application of API data fetching concepts learned in Lesson 12.

Techniques Covered:

- **API URLs:**
  - Defines URLs for fetching movie data and images.
- **Initial Display:**
  - Invokes `**showMovies**` with the initial API URL to display popular movies.
- **Show Movies Function:**
  - Fetches movie data from the API.
  - Dynamically creates HTML elements for each movie.
  - Appends movie info to the main container.
- **Search Event:**
  - Adds submit event to the form.
  - Prevents default form behavior.
  - Clears the main container for new results.
- **Search API Request:**
  - Constructs search API URL from user input.
  - Calls `**showMovies**` for search results.
  - Clears search input after submission.
- **API Key:**
  - Includes an API key for authentication in the API URLs.

Week 12 offers students a range of advanced projects, covering seats booking, user authentication, and API data fetching. The inclusion of API-related projects and the first home assignment further enhances their skills in data retrieval and application development.

Week 13: Culmination of JavaScript Learning

In the final week of your JavaScript journey, we'll cap off your learning with a series of practical challenges and a creative project that demonstrates your newfound skills.

Day 61:

55th Project: Find Meal

Students will engage in a practical project focused on finding meal-related information.

- **Event Listeners:**
  - Listens for form submission and button clicks.
  - Targets submit, random, and **mealsEl** elements.

- **Search Meal Function:**
  - Prevents default form submission.
  - Fetches meal data based on the search term.
  - Updates HTML with search results or displays a message.
- **Get Meal by ID Function:**
  - Fetches meal details by ID.
  - Calls **`addMealToDOM`** to display details on the page.
- **Get Random Meal Function:**
  - Clears existing meals and headings.
  - Fetches a random meal from the API.
  - Calls **`addMealToDOM`** to display the random meal.
- **Add Meal to DOM Function:**
  - Displays meal details on the page.
  - Creates HTML elements for name, image, category, area, instructions, and ingredients.
- **Event Delegation:**
  - Uses event delegation on **mealsEl**.
  - Handles clicks on meal items.
  - Retrieves the clicked meal's ID and displays its details using **`getMealById`**.
- **.map() Method:**
  - Generates HTML markup for each fetched meal.
  - Constructs HTML elements for displaying meal information.
- **.join() Method:**
  - Joins generated HTML markup into a single string.
  - Renders the string in a specific webpage container.
- **.push() Method:**
  - Dynamically populates the ingredients array with meal details.
- **.find() Method:**
  - Searches and returns the first occurrence in an array.
  - Identifies clicked meal information in the webpage path.
- **.forEach() Method:**
  - Iterates through the fetched meal array.
  - Constructs HTML elements for each meal display.

Day 62:

56th Project: Pokemons API

Students will dive into the Poke Mons API project, applying their skills to interact with this data source.

- Asynchronously fetches data for multiple Pokémon from the PokeAPI.
- Dynamically creates Pokémon containers and displays information like name, ID, and type.
- Applies background colors to containers based on the Pokémon's primary type.
- Utilizes string manipulation to format name, ID, and type.

- Demonstrates the use of loops to iterate through a range of Pokémon IDs.
- Utilizes `**map**` and `**find**` methods to extract and manipulate data.
- Showcases DOM manipulation by creating and styling dynamic elements.

Day 63:

Revision and Interview Preparation

Dedicate this day to revising key JavaScript concepts and solving interview-style questions to strengthen your skills.

Day 64-65:

57th Project: Creative Project

Students will unleash their creativity and demonstrate mastery by undertaking a personalized project, whether it's crafting a portfolio, designing a game, developing a blog, or bringing any imaginative idea to fruition.

This hands-on experience serves as a canvas for applying their JavaScript skills, allowing them to showcase their unique capabilities. This project marks the culmination of their journey in mastering this versatile programming language, providing a fitting and celebratory conclusion to their JavaScript learning adventure. Congratulations on reaching this milestone!