

AMALITCH

Week 1.4

Lab Activity: JavaScript Hero: Time Master

Learning Objectives:

Master the creation, manipulation, and interaction of JavaScript objects to build a dynamic and customizable clock application.

Introduction:

In this lab, you'll transform into a JavaScript Time Master, manipulating time itself by crafting a clock using the power of objects. You'll build a clock from the ground up, handling time data, formatting, and visual display, while reinforcing key object-oriented programming principles.

Challenges:

1. Time Object Basics:

- **Task:** Create a JavaScript `Date` object to represent the current time.
- **Evaluation:**
 - Code Review: Ensure proper use of the `Date` constructor and methods like `getHours()`, `getMinutes()`, and `getSeconds()`.
 - Output Test: Verify the ability to extract and log the current hour, minute, and second.

2. Object-Oriented Clock:

- **Task:** Design a `Clock` object with properties like hours, minutes, and seconds.
- **Evaluation:**
 - Code Review: Assess the correct use of properties and potential use of a constructor function for clock creation.
 - Output Test: Confirm the ability to create clock instances and access their properties.

3. Time Formatting:

- **Task:** Add methods to the `Clock` object for formatting time:
 - `getFormattedTime()` : Returns a string in the format "HH:MM:SS".
 - `get12HourTime()` : Returns a string with AM/PM.

- **Evaluation:**

- Code Review: Check for accurate time formatting logic within the methods.
- Output Test: Verify that formatted time strings are produced correctly.

4. Dynamic Display:

- **Task:** Create a function to display the clock on a webpage (e.g., using a div element).

- **Evaluation:**

- Code Review: Assess the use of DOM manipulation to update the clock display.
- Output Test: Confirm that the clock display updates every second with the correct time.

5. Clock Customization:

- **Task:** Add options to customize the clock (e.g., 12/24-hour format, time zone, colors).

- **Evaluation:**

- Code Review: Evaluate the implementation of customization options within the Clock object.
- Output Test: Verify that customization options affect the clock display as expected.

6. Advanced Challenge (Optional):

- **Task:** Build an alarm clock feature that triggers an alert at a specified time.

- **Evaluation:**

- Functionality: Assess the alarm's ability to trigger at the correct time and the user's ability to set the alarm.
- Code Review: Evaluate the code for the alarm functionality, including time comparison and alert mechanisms.

○