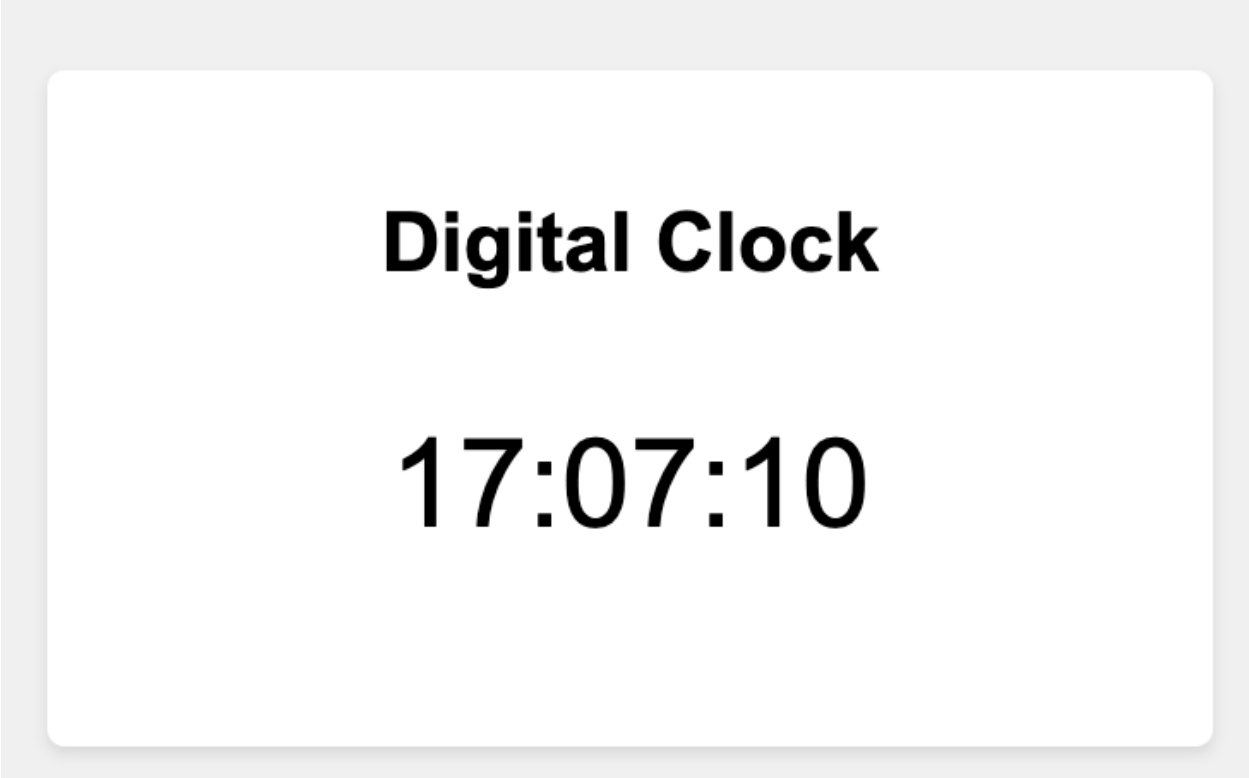
JavaScript code Example Create a Digital Clock



Project: Digital Clock

Step 1: HTML Structure

Create an HTML file named index.html and set up the basic structure.

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Digital Clock</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Digital Clock</h1>
        <p id="time"></p>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

Step 2: CSS Styling

Create a CSS file named styles.css for basic styling.

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    margin: 0;
```

```
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

.container {
    background-color: #fff;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    width: 300px;
    text-align: center;
}

h1 {
    font-size: 24px;
}

#time {
    font-size: 36px;
}
```

Step 3: JavaScript Logic

Create a JavaScript file named script.js for the application logic.

```
const timeElement = document.getElementById("time");

function updateTime() {
    const now = new Date();
    const hours = now.getHours().toString().padStart(2,
"0");
    const minutes =
now.getMinutes().toString().padStart(2, "0");
    const seconds =
now.getSeconds().toString().padStart(2, "0");
    const timeString =
`${hours}:${minutes}:${seconds}`;
    timeElement.textContent = timeString;
}

// Call updateTime every second (1000 milliseconds)
setInterval(updateTime, 1000);
```

```
// Initial call to display the time immediately
updateTime();
```

Step 4: Testing

Open the index.html file in a web browser. You should see a digital clock displaying the current time, updating every second.

Congratulations! You've successfully created a simple Digital Clock using HTML, CSS, and JavaScript. This project demonstrates how to manipulate time and update content dynamically on a webpage.

Here's the detailed breakdown of the JavaScript code:

We start by selecting the HTML element where we'll display the time using const timeElement = document.getElementById("time");.

We define the updateTime function which:

- Gets the current date and time using new Date().
- Extracts the hours, minutes, and seconds components and formats them with leading zeros using .padStart() method.
- Constructs a time string in the format "HH:MM:SS".
- Updates the content of timeElement with the constructed time string.

We use setInterval(updateTime, 1000); to call the updateTime function every 1000 milliseconds (1 second). This ensures that the clock updates every second.

We also call updateTime(); initially to immediately display the current time when the page loads.

Feel free to explore and expand this project further by adding features like displaying the current date, customizing the clock's appearance, or even implementing time zones.

```
// Get reference to the HTML element where the time
will be displayed
const timeElement = document.getElementById("time");

// Define the function to update the time
function updateTime() {
    // Create a new Date object to get the current time
    const now = new Date();

    // Extract hours, minutes, and seconds from the
Date object
    const hours = now.getHours().toString().padStart(2,
"0");
    const minutes =
now.getMinutes().toString().padStart(2, "0");
    const seconds =
now.getSeconds().toString().padStart(2, "0");
```

```
    // Create a formatted time string in "HH:MM:SS"
format
    const timeString =
`${hours}:${minutes}:${seconds}`;

    // Update the text content of the timeElement with
the new time string
    timeElement.textContent = timeString;
}

// Call updateTime every second (1000 milliseconds) to
keep the clock updated
setInterval(updateTime, 1000);

// Initial call to updateTime to display the time
immediately when the page loads
updateTime();
```

Step by Step Explanation:

1. We start by getting a reference to the HTML element where we want to display the time. In this case, it's the timeElement obtained using document.getElementById("time").

2. We define the updateTime function which is responsible for updating the displayed time.
3. Inside the updateTime function:
    a. We create a new Date object called now to capture the current date and time.
4. We extract the hours, minutes, and seconds components from the now object using the getHours(), getMinutes(), and getSeconds() methods. We use the .toString().padStart(2, "0") chain to ensure that single-digit values are formatted with a leading zero.
5. We construct a formatted time string using template literals (backticks) with the hours, minutes, and seconds components.
6. We update the text content of the timeElement with the newly formatted time string using timeElement.textContent = timeString.
7. We use the setInterval(updateTime, 1000) function to call the updateTime function every 1000 milliseconds (1 second). This ensures that the displayed time updates dynamically every second.
8. Finally, we call updateTime() initially to immediately display the current time when the page loads.

By following these steps, the code creates a simple digital clock that continuously updates with the current time. This project demonstrates how to use JavaScript to interact with date and time objects, format them, and dynamically update content on a webpage.