JavaScript JS + CSS Clock Explanation

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

<head>
  <meta charset="UTF-8">
  <title>JS + CSS Clock</title>
  <link rel="icon" href="https://fav.farm/✍">
  <link rel="stylesheet" href="style.css">

</head>

<body>

  <div class="clock">
    <div class="clock-face">
      <div class="hand hour-hand"></div>
      <div class="hand min-hand"></div>
      <div class="hand second-hand"></div>
    </div>
  </div>

  <div class="stopwatch">
    <p id="stopwatch-display">00:00:00</p>
    <div class="stopwatchButtons">
    <button id="startStopwatch">Start</button>
    <button id="resetStopwatch">Reset</button>
  </div>
  </div>

  <script>

    // Code made for clock with Tutorial

    const secondHand = document.querySelector('.second-hand');
    const minsHand = document.querySelector('.min-hand');
    const hourHand = document.querySelector('.hour-hand');

    function setDate() {
      const now = new Date();

      const seconds = now.getSeconds();
      const secondsDegrees = ((seconds / 60) * 360) + 90;
      secondHand.style.transform = `rotate(${secondsDegrees}deg)`;
```

```
      const mins = now.getMinutes();
      const minsDegrees = ((mins / 60) * 360) + ((seconds / 60) * 6) +
90;
      minsHand.style.transform = `rotate(${minsDegrees}deg)`;

      const hour = now.getHours();
      const hourDegrees = ((hour / 12) * 360) + ((mins / 60) * 30) +
90;
      hourHand.style.transform = `rotate(${hourDegrees}deg)`;
    }

    setInterval(setDate, 1000);

    setDate();
</script>
```

Below is a brief explanation of how the JS+CSS Clock is built:

In the setDate() function, the code I wrote based on a tutorial is used to update the clock hands based on the current time. Here's an overview of what happens in the function:

Firstly, the clock hands elements (seconds, minutes, and hours) are selected using the document.querySelector() method. These elements are stored in variables called secondHand, minsHand, and hourHand. Then, the new Date() constructor is used to obtain the current time and date. The time values (seconds, minutes, and hours) are extracted from the obtained timestamp using the getSeconds(), getMinutes(), and getHours() methods of the Date object.

Based on these time values, the rotation angles are calculated for each hand. For example, for the second hand (secondHand), the rotation angle is calculated by taking the ratio of the current seconds to 60 (the number of seconds in a minute) and multiplying it by 360 degrees. A 90-degree offset is also added to align the hands correctly with the 12 o'clock mark.

The transform property of the clock hands is set using the calculated rotation angles, which visually updates the hands on the web page.

To ensure that the clock hands are continuously updated, the setDate() function is called again every second using the setInterval() function. This provides a smooth movement of the hands and keeps the time up to date.

What I have learned

From this challenge, I have learned how to create a JavaScript and CSS-powered clock.
By understanding and utilizing the JavaScript Date object, I have gained knowledge on retrieving the current time and converting it into rotational degrees for the clock hands.

Through CSS transformations and the manipulation of the transform property in JavaScript, I have acquired the ability to visually update the clock hands' positions to represent the current time accurately.
Overall, this challenge has provided me with valuable insights into combining JavaScript and CSS to create interactive and functional elements on a web page.