Lesson : Timers

Timers

Timeouts

Some events are related to time instead of user actions. Ex: A website may wish to display an advertisement 10 seconds after the webpage loads or display inventory data that updates at regular intervals. A timer is a general name for techniques to execute JavaScript code after some amount of time has occurred.

A web browser is able to execute a function after a time delay using `setTimeout()`.
The setTimeout() method takes two arguments: a function and a time delay in milliseconds (1/1000th of a second). The browser calls the function after the time delay. `setTimeout()` returns a unique integer identifier that refers to the timeout that was created, and the timeout can be canceled by passing the identifier to clearTimeout().

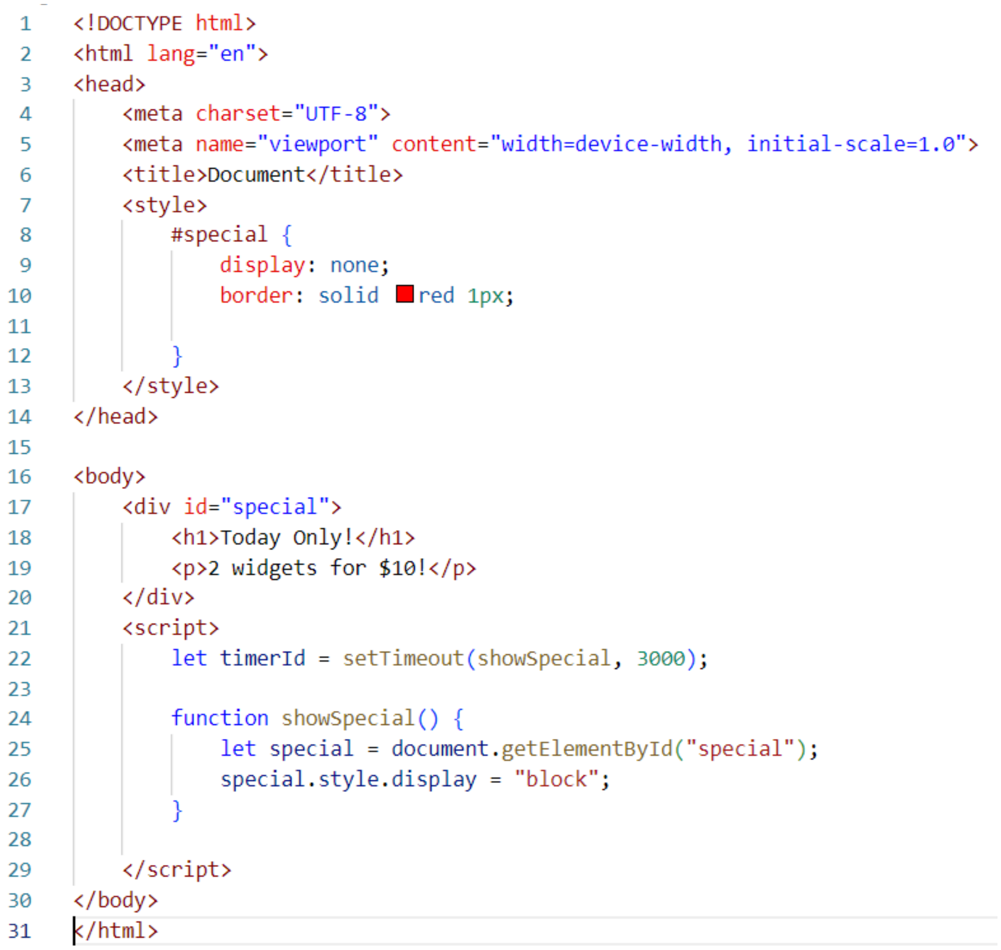
click on the following link and execute to understand:

https://replit.com/join/mtetaajntw-pragaticoder (https://replit.com/join/mtetaajntw-pragaticoder)

Explanation :

1. A webpage contains a <div> with a daily special that is not yet visible.
2. setTimeout() tells the browser to call showSpecial() in 3 seconds.
3. After 3 seconds, the browser calls showSpecial() and makes the <div> visible.

Example :



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        #special {
            display: none;
            border: solid red 1px;

        }
    </style>
</head>

<body>
    <div id="special">
        <h1>Today Only!</h1>
        <p>2 widgets for $10!</p>
    </div>
    <script>
        let timerId = setTimeout(showSpecial, 3000);

        function showSpecial() {
            let special = document.getElementById("special");
            special.style.display = "block";
        }

    </script>
</body>
</html>
```

Intervals

A web browser is able to execute a function repeatedly with a time delay between calls using `setInterval()`. The setInterval() method takes two arguments: a function and a time interval in milliseconds (*t*). The browser calls the function every *t* milliseconds until the interval is canceled. The `setInterval()` method returns the interval's unique integer identifier, and the interval identifier can be passed to the clearInterval() method to cancel the interval.

Example :

Following program implements the setInterval function.

Copy and Paste the code, Run, and check :

1. A webpage contains a <div> with a daily special that is not yet visible.
2. setTimeout() tells the browser to call showSpecial() in 3 seconds.
3. After 3 seconds, the browser calls showSpecial() and makes the <div> visible.

<!DOCTYPE html>

<html lang="en">

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
        <img src="ball2.png" id="ball" alt="ball" style="position: absolute; left: 0px; top: 0px;" style="position: absolute; left: 0px; top: 0px;">
        <input type ="button" onclick="startMoving()" value="Start Moving" style="position: absolute; left: 20px; top: 60px;color:green"></button>

        <script>
            let ballImage;
            let timerId;

            function startMoving() {
                ballImage = document.getElementById("ball");
                timerId = setInterval(moveBall, 10);
            }

            function moveBall() {
                let left = parseInt(ballImage.style.left);
                ballImage.style.left = left + 5 + "px";
            }
        </script>
    </body>
    </html>

</body>
</html>
```