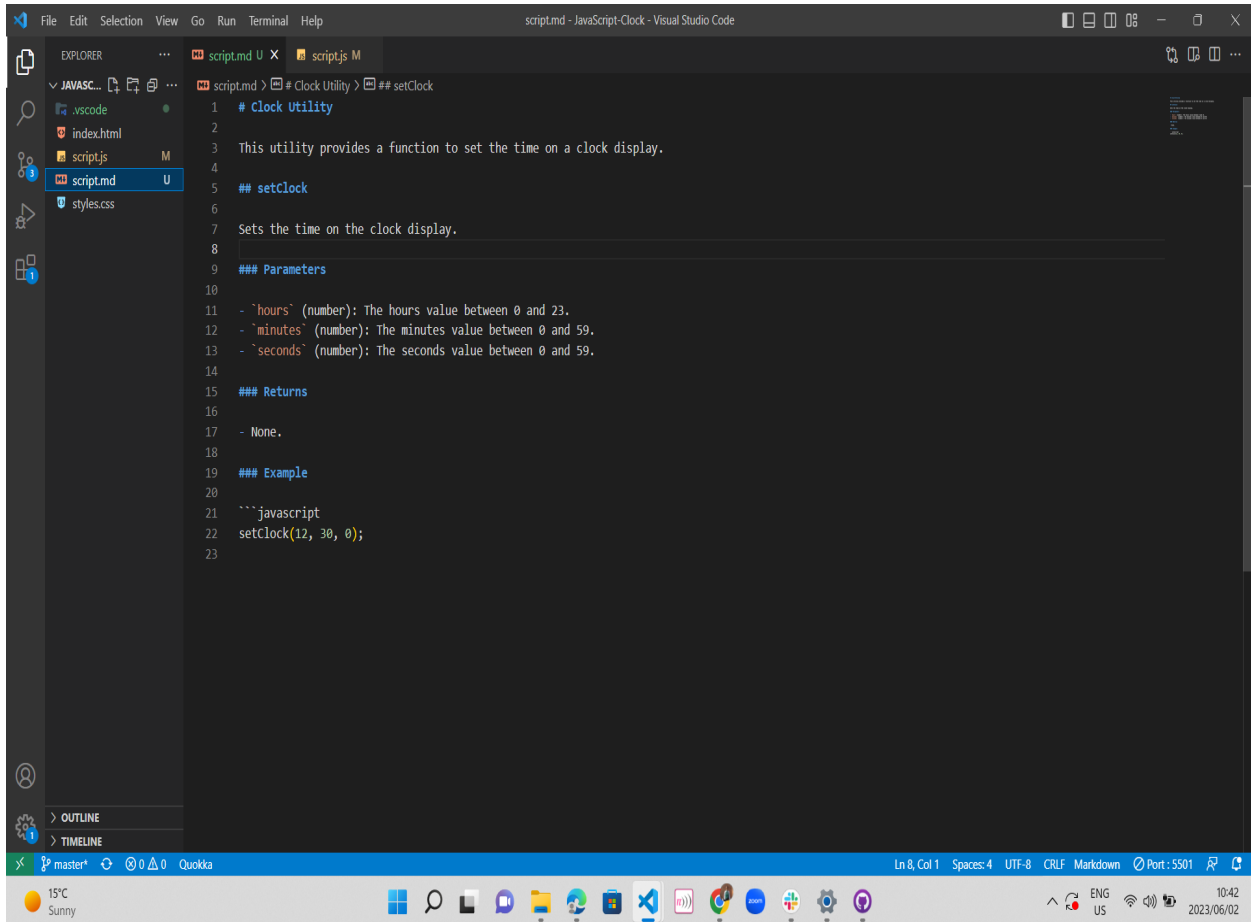


DWA_03.4 Knowledge Check_DWA3.1

1. Please show how you applied a Markdown File to a piece of your code.



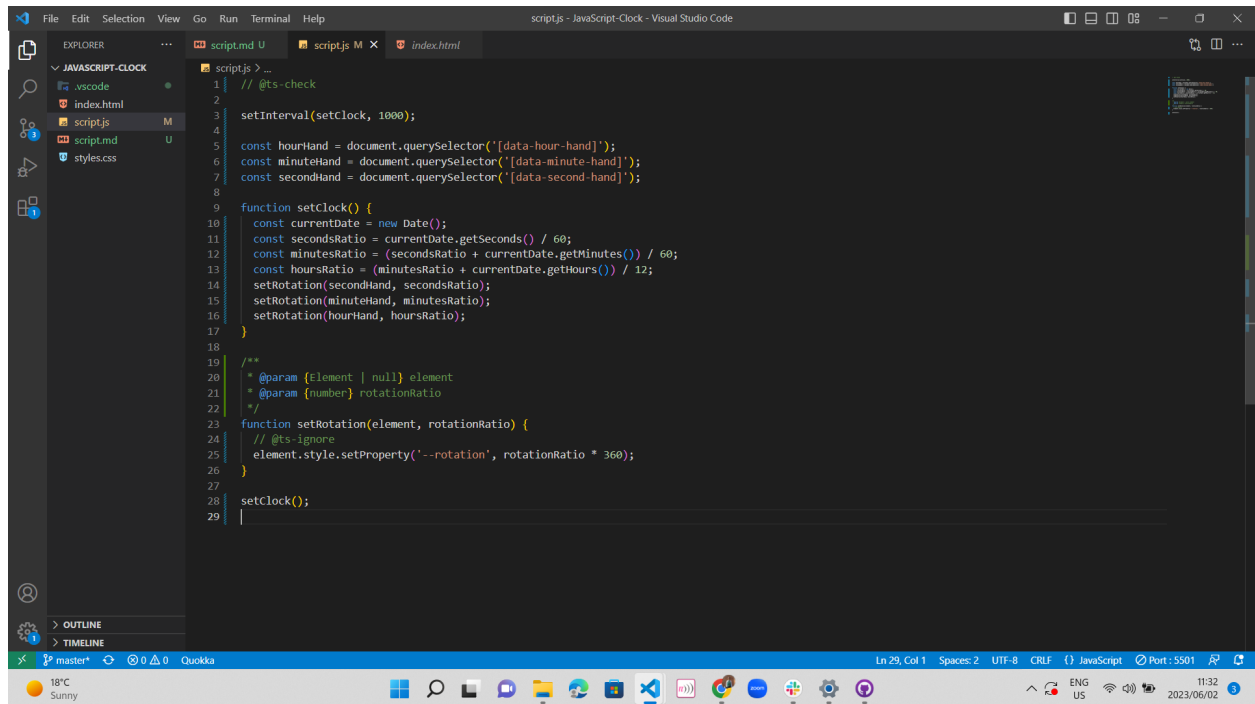
The screenshot shows the Visual Studio Code editor with a file named `script.md` open. The file contains a JavaScript utility function `setClock` with JSDoc comments. The comments are as follows:

```
1  # clock utility
2
3  This utility provides a function to set the time on a clock display.
4
5  ## setClock
6
7  Sets the time on the clock display.
8
9  ### Parameters
10
11  - `hours` (number): The hours value between 0 and 23.
12  - `minutes` (number): The minutes value between 0 and 59.
13  - `seconds` (number): The seconds value between 0 and 59.
14
15  ### Returns
16
17  - None.
18
19  ### Example
20
21  ```javascript
22  setClock(12, 30, 0);
23
```

2. Please show how you applied JSDoc Comments to a piece of your code.

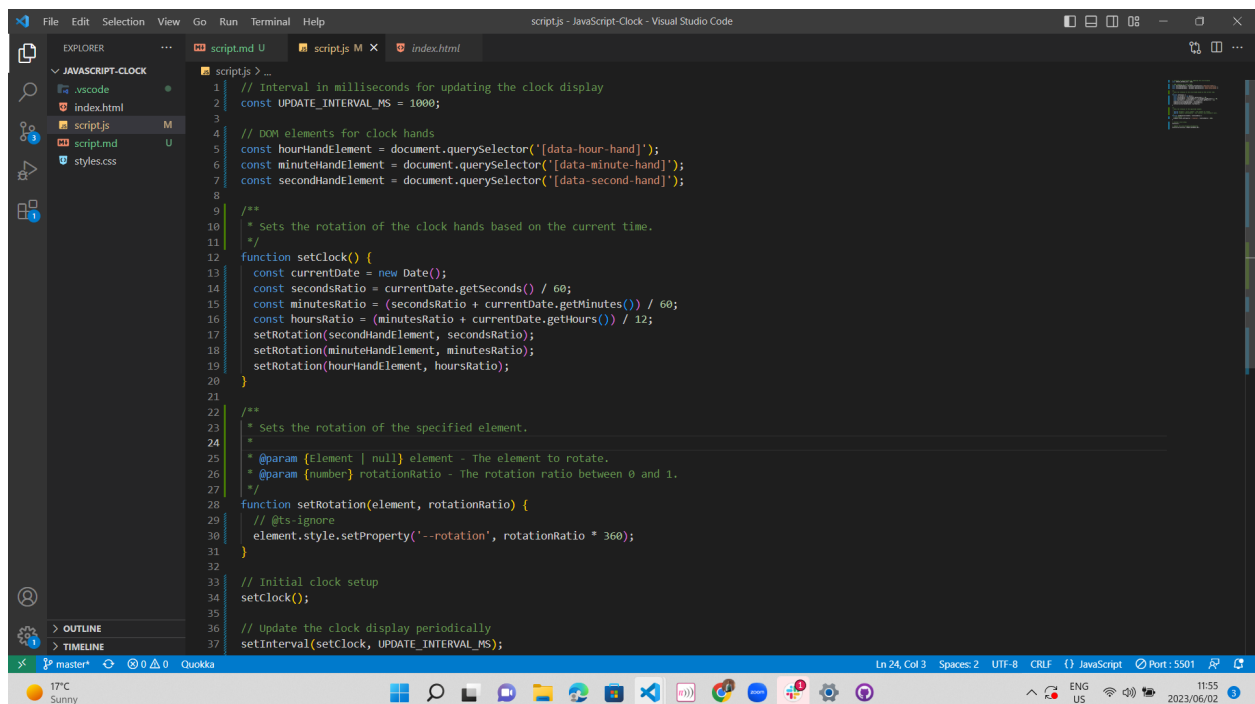
```
1 /**
2  * Sets the clock display by updating the rotation of the clock hands.
3  * Uses the current time to calculate the rotation ratios.
4  */
5  function setClock() {
6      const currentDate = new Date();
7      const secondsRatio = currentDate.getSeconds() / 60;
8      const minutesRatio = (secondsRatio + currentDate.getMinutes()) / 60;
9      const hoursRatio = (minutesRatio + currentDate.getHours()) / 12;
10
11      setRotation(secondHand, secondsRatio);
12      setRotation(minuteHand, minutesRatio);
13      setRotation(hourHand, hoursRatio);
14  }
15
16  /**
17   * Sets the rotation of the given element using the provided rotation ratio.
18   *
19   * @param {HTMLElement} element - The element to rotate.
20   * @param {number} rotationRatio - The rotation ratio between 0 and 1.
21   */
22  function setRotation(element, rotationRatio) {
23      element.style.setProperty('--rotation', rotationRatio * 360);
24  }
25
26  // Select clock hands
27  const hourHand = document.querySelector('[data-hour-hand]');
28  const minuteHand = document.querySelector('[data-minute-hand]');
29  const secondHand = document.querySelector('[data-second-hand]');
30
31  // Set the clock initially
32  setClock();
33
34  // Update the clock every second
35  setInterval(setClock, 1000);
36
```

3. Please show how you applied the @ts-check annotation to a piece of your code.



```
1 // @ts-check
2
3 setInterval(setClock, 1000);
4
5 const hourHand = document.querySelector('[data-hour-hand]');
6 const minuteHand = document.querySelector('[data-minute-hand]');
7 const secondHand = document.querySelector('[data-second-hand]');
8
9 function setClock() {
10   const currentDate = new Date();
11   const secondsRatio = currentDate.getSeconds() / 60;
12   const minutesRatio = (secondsRatio + currentDate.getMinutes()) / 60;
13   const hoursRatio = (minutesRatio + currentDate.getHours()) / 12;
14   setRotation(secondHand, secondsRatio);
15   setRotation(minuteHand, minutesRatio);
16   setRotation(hourHand, hoursRatio);
17 }
18
19 /**
20  * @param {element | null} element
21  * @param {number} rotationRatio
22  */
23 function setRotation(element, rotationRatio) {
24   // @ts-ignore
25   element.style.setProperty('--rotation', rotationRatio * 360);
26 }
27
28 setClock();
29
```

4. As a BONUS, please show how you applied any other concept covered in the 'Documentation' module.



```
1 // Interval in milliseconds for updating the clock display
2 const UPDATE_INTERVAL_MS = 1000;
3
4 // DOM elements for clock hands
5 const hourHandElement = document.querySelector('[data-hour-hand]');
6 const minuteHandElement = document.querySelector('[data-minute-hand]');
7 const secondHandElement = document.querySelector('[data-second-hand]');
8
9 /**
10  * Sets the rotation of the clock hands based on the current time.
11  */
12 function setClock() {
13   const currentDate = new Date();
14   const secondsRatio = currentDate.getSeconds() / 60;
15   const minutesRatio = (secondsRatio + currentDate.getMinutes()) / 60;
16   const hoursRatio = (minutesRatio + currentDate.getHours()) / 12;
17   setRotation(secondHandElement, secondsRatio);
18   setRotation(minuteHandElement, minutesRatio);
19   setRotation(hourHandElement, hoursRatio);
20 }
21
22 /**
23  * Sets the rotation of the specified element.
24  *
25  * @param {element | null} element - The element to rotate.
26  * @param {number} rotationRatio - The rotation ratio between 0 and 1.
27  */
28 function setRotation(element, rotationRatio) {
29   // @ts-ignore
30   element.style.setProperty('--rotation', rotationRatio * 360);
31 }
32
33 // Initial clock setup
34 setClock();
35
36 // Update the clock display periodically
37 setInterval(setClock, UPDATE_INTERVAL_MS);
38
```
