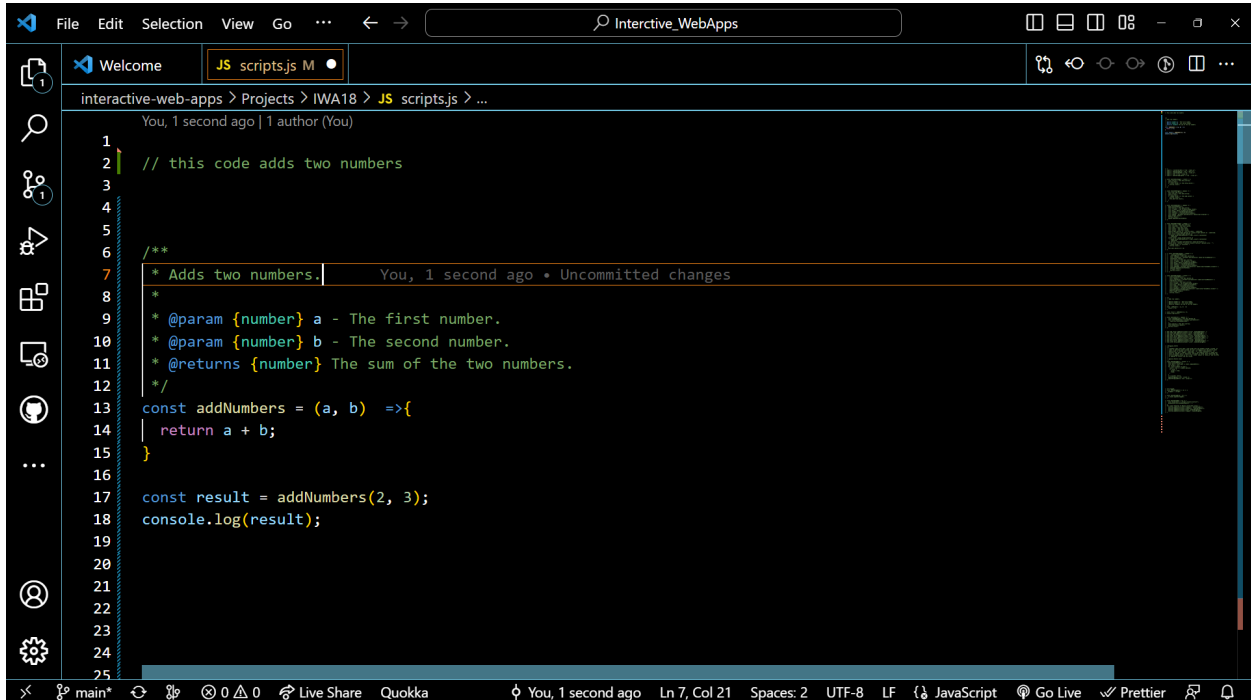


DWA_03.4 Knowledge Check_DWA3.1

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



The screenshot shows a VS Code editor window with a file named `scripts.js` open. The code is a JavaScript function `addNumbers` that takes two parameters, `a` and `b`, and returns their sum. The function is called with `addNumbers(2, 3)` and the result is logged to the console. The code is formatted with Prettier. A JSDoc comment is applied to the `addNumbers` function, describing its purpose and parameters. The comment is written in a multi-line format, with each line starting with an asterisk and a space. The comment describes the function's purpose, the parameters, and the return value. The status bar at the bottom shows the file is in the `main` branch, the editor is in `JavaScript` mode, and the Prettier formatter is active.

```
1 // this code adds two numbers
2
3
4
5
6 /**
7  * Adds two numbers.
8  *
9  * @param {number} a - The first number.
10 * @param {number} b - The second number.
11 * @returns {number} The sum of the two numbers.
12 */
13 const addNumbers = (a, b) =>{
14   return a + b;
15 }
16
17 const result = addNumbers(2, 3);
18 console.log(result);
19
20
21
22
23
24
25
```

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

The screenshot shows the VS Code editor interface. The top bar indicates the file is 'JS scripts.js M'. The breadcrumb trail is 'interactive-web-apps > Projects > IWA18 > JS scripts.js > addNumbers'. The code in the editor is as follows:

```
100
101  /**
102   * Adds two numbers.
103   *
104   * @param {number} a - The first number.
105   * @param {number} b - The second number.
106   * @returns {number} The sum of the two numbers.
107   */
108  const addNumbers = (a, b) =>{
109    return a + b;
110  }
111
112  const result = addNumbers(2, 3);
113  console.log(result);
114
115
```

A status bar at the bottom shows 'main*' and 'Live Share Quokka'. The status bar at the very bottom indicates 'LF', 'JavaScript', 'Go Live', and 'Prettier'.

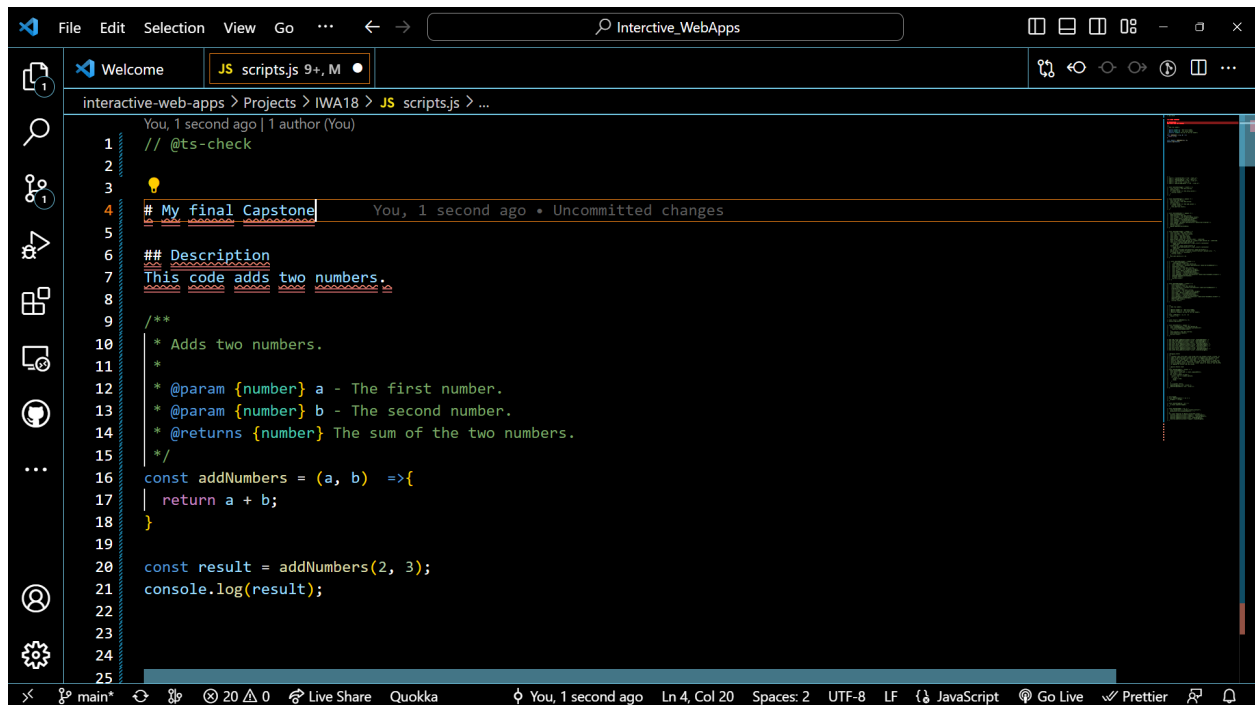
3. Please show how you applied the @ts-check annotation to a piece of your code.

The screenshot shows the same VS Code editor interface, but with the '@ts-check' annotation added at the top of the file. The code is now as follows:

```
1  // @ts-check
2
3
4
5  /**
6   * Adds two numbers.
7   *
8   * @param {number} a - The first number.
9   * @param {number} b - The second number.
10  * @returns {number} The sum of the two numbers.
11  */
12  const addNumbers = (a, b) =>{
13    return a + b;
14  }
15
16  const result = addNumbers(2, 3);
17  console.log(result);
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
```

The status bar at the bottom now shows 'main*' and 'Live Share Quokka'. The status bar at the very bottom indicates 'Ln 20, Col 1', 'Spaces: 2', 'UTF-8', 'LF', 'JavaScript', 'Go Live', and 'Prettier'.

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



The screenshot shows a VS Code editor window titled 'Interactive_WebApps'. The editor is open to a file named 'scripts.js'. The code in the file is as follows:

```
1 // @ts-check
2
3
4 # My final Capstone You, 1 second ago • Uncommitted changes
5
6 ## Description
7 This code adds two numbers.
8
9 /**
10  * Adds two numbers.
11  *
12  * @param {number} a - The first number.
13  * @param {number} b - The second number.
14  * @returns {number} The sum of the two numbers.
15  */
16 const addNumbers = (a, b) =>{
17   | return a + b;
18 }
19
20 const result = addNumbers(2, 3);
21 console.log(result);
22
23
24
25
```

The code includes a JSDoc comment with a title, description, and parameters. The function 'addNumbers' is defined and called with arguments 2 and 3. The result is logged to the console.