

DWA_03.4 Knowledge Check_DWA3.1

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

IWA1 >  README.md > ...

```
1
2 # Tally counter
3
4 This is a simple tally counter web page that allows you to increment and decrement a displayed value. The counter has a minimum and maximum limit, and the corresponding
  buttons are disabled when the limits are reached.
5
6
7 ## Features
8
9 - Increment and Decrement: The app allows users to increase or decrease the displayed value of the counter by clicking on the respective buttons.
10 - Value Limits: The counter has predefined limits to prevent the value from going beyond a specified range. The MAX_NUMBER constant defines the maximum value, and the
  MIN_NUMBER constant defines the minimum value.
11 - Disabled Buttons: When the counter reaches the maximum or minimum limit, the corresponding increment or decrement button is disabled to prevent further incrementing or
  decrementing.
12 - Counter Selection: The app includes a "Counter" dropdown menu that allows users to select different counter options. However, the functionality for handling this option is
  not implemented in the provided JavaScript code.
13
```

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

```
/**
 * This function decreases the {@link newValue} by 1 until it reaches the {@link MIN_NUMBER}
 * When the {@link MIN_NUMBER} is reached the {@link subtract} button is disabled
 * @returns {void}
 */

const subtractHandler = () => {

  /**
   * @type {number} newValue- takes the string value that was fetched from the DOM using the querySelector and converts it into a number
   */
  const newValue = parseInt(number.Value) - 1;
  number.value = newValue;

  if (add.disabled === true) {
    add.disabled = false;
  }

  if (newValue <= MIN_NUMBER) {
    subtract.disabled = true;
  }
}
```

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

```
/**
 * @type {number} newValue- takes the string value that was fetched from the DOM using the querySelector and converts it into a number type
 */
const newValue = parseInt(number.Value) + '1';
number.value = newValue;
```

Type 'string' is not assignable to type 'number'. ts(2322)

const newValue: number

@type — {number} newValue- takes the string value that was fetched from the DOM using the querySelector and converts it into a number type

[View Problem \(Alt+F8\)](#) [Quick Fix... \(Ctrl+.\)](#)

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

```
IWA14 > js challenge2.js > add
1  /**
2   * This function takes parameter 'a' and parameter 'b' and adds them together and returns a number
3   *
4   * @param {number} a
5   * @param {number} b
6   * @returns {number}
7   */
8
9  function add(a, b) {
10
11      return a + b
12
13  }
14
```