

# DWA\_03.4 Knowledge Check\_DWA3.1

---

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

## IWA 1 - Tally Count

```
DWA-3 > example.md > # Tally App > ## Usage
1  # Tally App
2
3  The Tally App is a simple tool to keep track of counts. It allows you to increment or decrement the count and provides a convenient way to
   visualize and manage your tallies.
4
5  Table of Contents (out of date)
6  - [Features](#features)
7  - [Usage](#usage)
8  - [Screenshots](#screenshots)
9
10 ## Features
11
12 - Create multiple tallies for different purposes.
13 - Increment or decrement the count for each tally.
14 - View the current count for each tally.
15 - Reset a tally to zero.
16 - Delete a tally.
17
18 ## Usage
19
20 To use the Tally App, follow these steps:
21
22 1. Open the Tally App on your device.
23 2. Create a new tally by clicking on the "New Tally" button.
24 3. Enter a name for the tally in the provided field and click "Create".
25 4. Once created, you will see the tally name and the current count displayed.
26 5. To increment the count, click on the "+" button next to the tally.
27 6. To decrement the count, click on the "-" button next to the tally.
28 7. If you want to reset the count to zero, click on the "Reset" button.
29 8. If you want to delete the tally, click on the "Delete" button.
```

---

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

## IWA 6 - Challenge 2

DWA-3 > JS jsdoc.js > ...

```
1  /**
2   * Represents financial calculations based on predefined variables.
3   */
4
5  /** @type {number} */
6  const rent = 400;
7
8  /** @type {string} */
9  const tax = '8%';
10
11 /** @type {number} */
12 const food = 51.7501;
13
14 /** @type {number} */
15 const salary = 800;
16
17 /** @type {number} */
18 const transport = 10.2;
19
20 /** @type {number} */
21 const hourOfDay = 0;
22
23 /** @type {number} */
24 const minuteOfDay = 0;
```

---

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

**IWA 6 - Challenge2**

```

// @ts-check

const rent = 400;
const tax = '8%';
const food = 51.7501;
const salary = 800;
const transport = 10.2;
const hourOfDay = 0;
const minuteOfDay = 0;

// Only change below this line

if (hourOfDay === 0 && minuteOfDay === 0) {
  const taxAsDecimal = parseFloat(tax) / 100;
  const startingAfterTax = salary - salary * taxAsDecimal;
  const balance = startingAfterTax - transport - food - rent;
  const currency = 'R'; // declared 'R'

  console.log(currency, balance.toFixed(2));
}

```

-With the `@ts-check` annotation, TypeScript will perform static type checking in the code and provide you with type-related warnings or errors during the development, helping you catch potential issues early.

---

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

---