

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

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

Here is an example markdown file of a README.MD for a calculator app

```
① README.MD > # Calculator App > ## Getting Started
1 # Calculator App
2
3 This app is a simple calculator designed to perform basic mathematical operations in the browser. It allows users to
  input numerical values and perform calculations such as addition, subtraction, multiplication, and division.
4
5 ## KeyFeatures
6 - User-friendly Interface: The calculator app provides a clean and intuitive interface...
7 - Basic Arithmetic Operations: Users can perform common arithmetic operations, including...
8
9
10 ## Requirements
11 - Browser like [Google Chrome](https://www.google.com/chrome/?brand=YTUH&
  gclid=Cj0KCQjwjryjBhD0ARIsAMLvnF-3L0TzodRcW_2RQRSpyC9P4trNGryWuA9kcur07xyQ0p_AAghUN3waAr2IEALw_wcB&gclsrc=aw.ds)
12 - Latest Version of Windows [Windows 11](https://www.microsoft.com/en-za/windows?r=1)
13
14 ## Getting Started
15 1. Open the app in your web browser.
16 2. Enter the numerical values you want to calculate using the provided buttons or the keyboard.
17 3. Select the desired arithmetic operation using the appropriate button (+, -, ×, ÷).
18
19
20 ## User Stories
21 - As a user, I want to perform basic arithmetic calculations such as addition, subtraction, multiplication, and
  division, so that I can quickly and accurately compute mathematical operations.
22 - As a user, I want the calculator app to provide a convenient way to calculate percentages, so that I can easily
  determine percentages of given values without manual calculations.
```

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

This is an example of a JSDoc comment that include @param and @return

```
script.js
1 /**
2  * This function multiplies two numbers together.
3  *
4  * @param {number} a - the first number
5  * @param {number} b - the second number
6  * @returns {number} The multiplication of these two numbers
7  */
8
9 function multiplyTheseNumbers(a,b) {
10   return a * b;
11 }
```

25

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

Because I set the type in the comment above, when I entered a string it indicated an error (left photo), but when I entered a number it didn't (right photo).

Indicates error

```
1 // @ts-check
2
3 /**
4  * Number of Tickets
5  * @type {number}
6  */
7
8 const numberOfTickets = two;
```

No error as I entered a number

```
1 // @ts-check
2
3 /**
4  * Number of Tickets
5  * @type {number}
6  */
7
8 const numberOfTickets = 2;
```

---

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

Example showing @typedef and @prop

```
3  /**
4   * guest
5   * @typedef {Object} Guest
6   * @property {number} id - Guest Id
7   * @property {string} name - Guest Name
8   * @property {number} age - Guest Age
9   * @property {boolean} isPresent - Guest is present
10  */
11
12  /**
13   * @type {Guest}
14   */
15  const guest = {
16    id: 7,
17    name: 'Anna',
18    age: 40,
19    isPresent: true
20  }
```

---