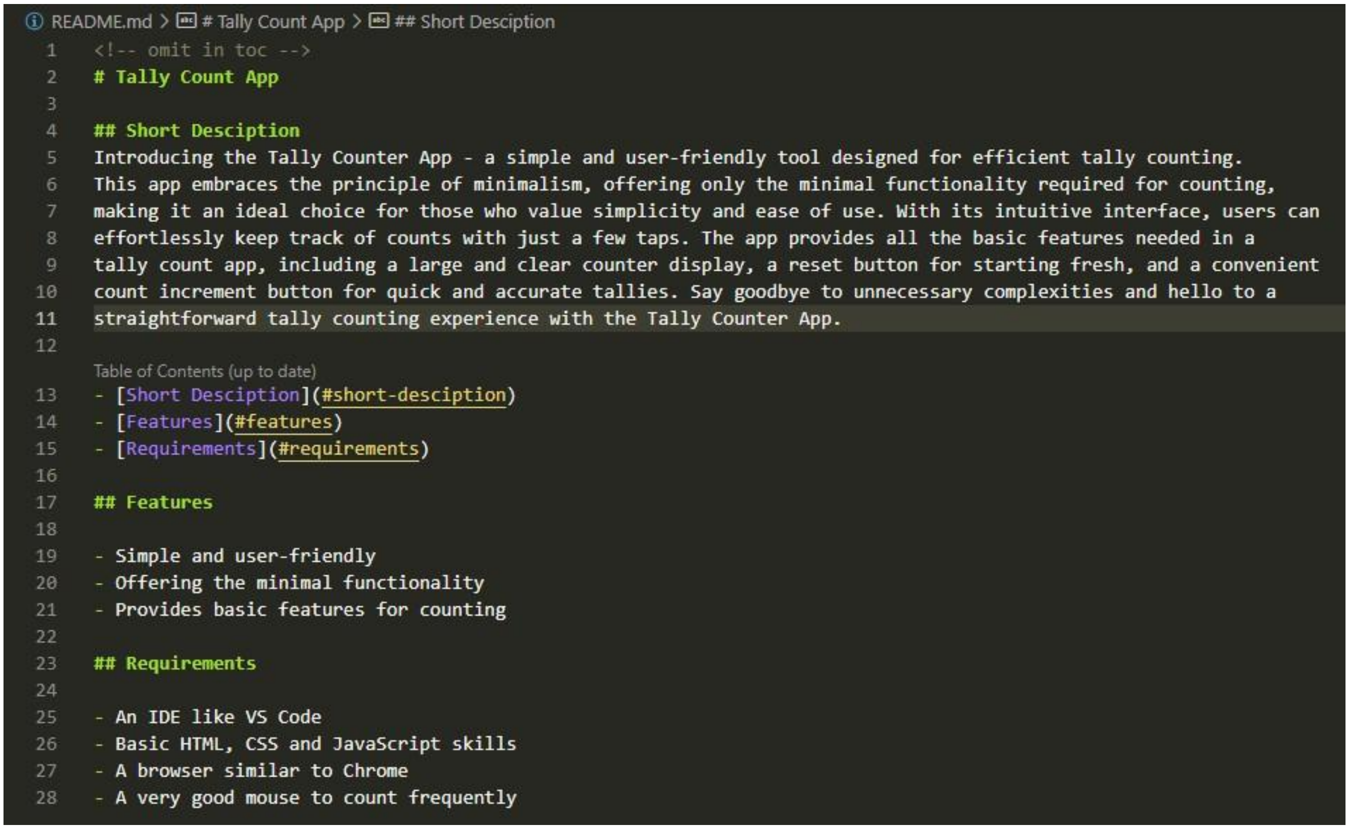
DWA_03.4 Knowledge Check_DWA3.1

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

Markdown is a lightweight markup language that describes how text should look on a page. Markdown is a style of writing documents that makes it easy to define what the content should look like. It describes headers, text styles, links, lists and so much more.

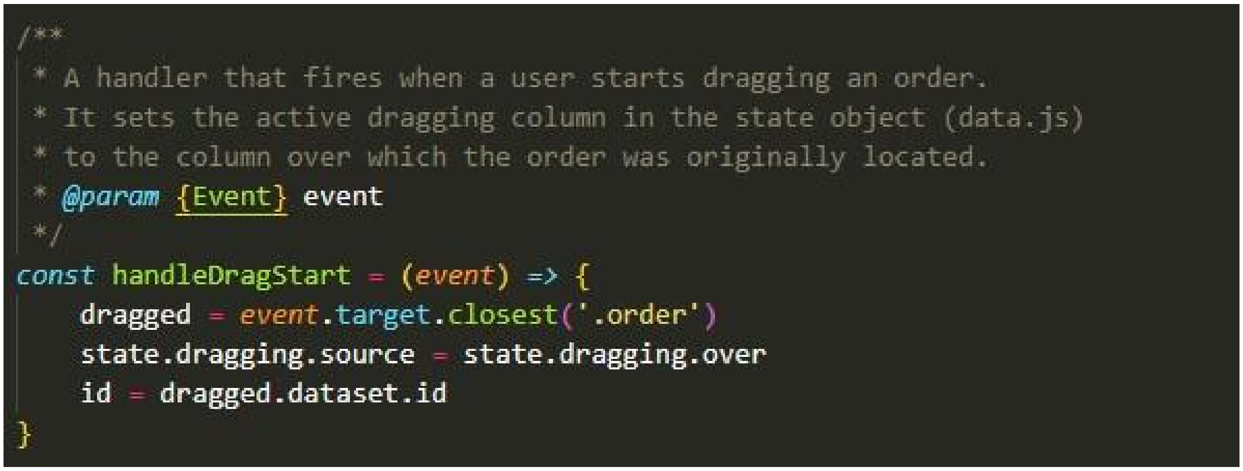
Markdown is used in documentation, articles, and notes and can even be used to build a webpage. If you use GitHub, you'll be familiar with the “readme.md” files that show up in the root of a repository. That “md” stands for markdown. You can even create a readme on your GitHub profile to customize your GitHub profile page.


README.md > # Tally Count App > ## Short Desciption
Table of Contents (up to date)

```markdown
<!-- omit in toc -->
# Tally Count App

## Short Desciption
Introducing the Tally Counter App - a simple and user-friendly tool designed for efficient tally counting.
This app embraces the principle of minimalism, offering only the minimal functionality required for counting,
making it an ideal choice for those who value simplicity and ease of use. With its intuitive interface, users can
effortlessly keep track of counts with just a few taps. The app provides all the basic features needed in a
tally count app, including a large and clear counter display, a reset button for starting fresh, and a convenient
count increment button for quick and accurate tallies. Say goodbye to unnecessary complexities and hello to a
straightforward tally counting experience with the Tally Counter App.

- [Short Desciption](#short-desciption)
- [Features](#features)
- [Requirements](#requirements)

## Features

- Simple and user-friendly
- Offering the minimal functionality
- Provides basic features for counting

## Requirements

- An IDE like VS Code
- Basic HTML, CSS and JavaScript skills
- A browser similar to Chrome
- A very good mouse to count frequently
```

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

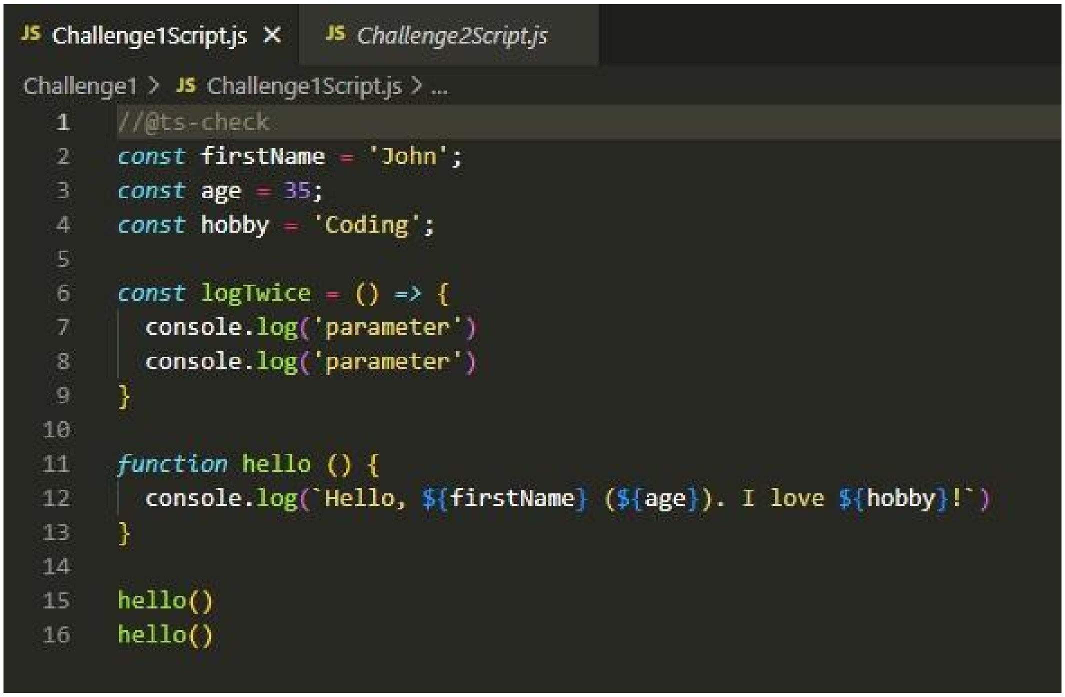
JSDoc is a markup language used to annotate JavaScript source code files. Using comments containing JSDoc, programmers can add documentation describing the application programming interface of the code they're creating.



```
/**
 * A handler that fires when a user starts dragging an order.
 * It sets the active dragging column in the state object (data.js)
 * to the column over which the order was originally located.
 * @param {Event} event
 */
const handleDragStart = (event) => {
    dragged = event.target.closest('.order')
    state.dragging.source = state.dragging.over
    id = dragged.dataset.id
}
```

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

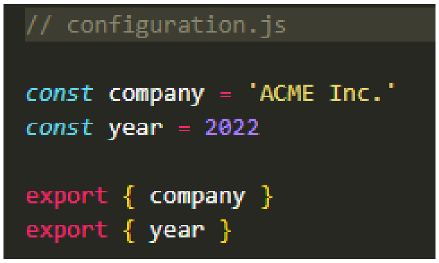
Adding @ts-check in your JavaScript files enables TypeScript to raise an error if there are any.



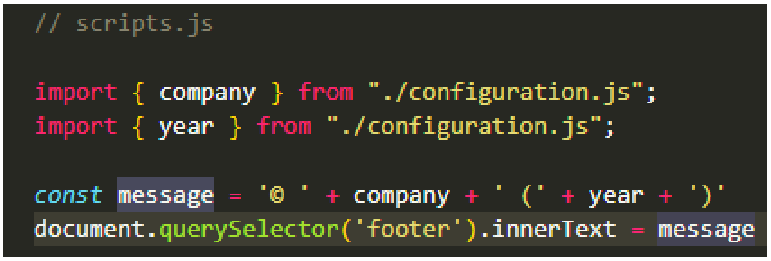
```
//@ts-check
const firstName = 'John';
const age = 35;
const hobby = 'Coding';

const logTwice = () => {
  console.log('parameter')
  console.log('parameter')
}

function hello () {
  console.log(`Hello, ${firstName} (${age}). I love ${hobby}!`)
}

hello()
hello()
```

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

Import and Export example of my IWA3 Challenges.



```
// configuration.js

const company = 'ACME Inc.'
const year = 2022

export { company }
export { year }
```



```
// scripts.js

import { company } from "./configuration.js";
import { year } from "./configuration.js";

const message = '© ' + company + ' (' + year + ')'
document.querySelector('footer').innerText = message
```