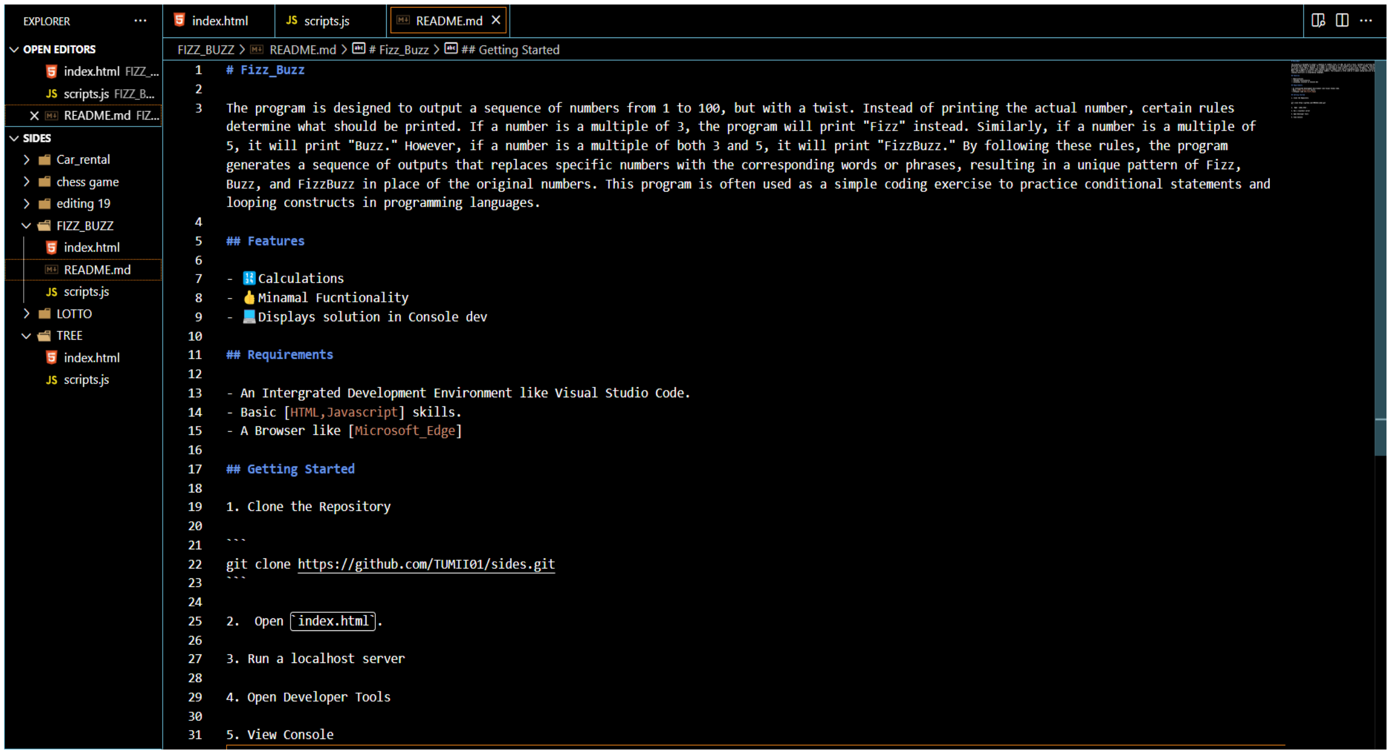
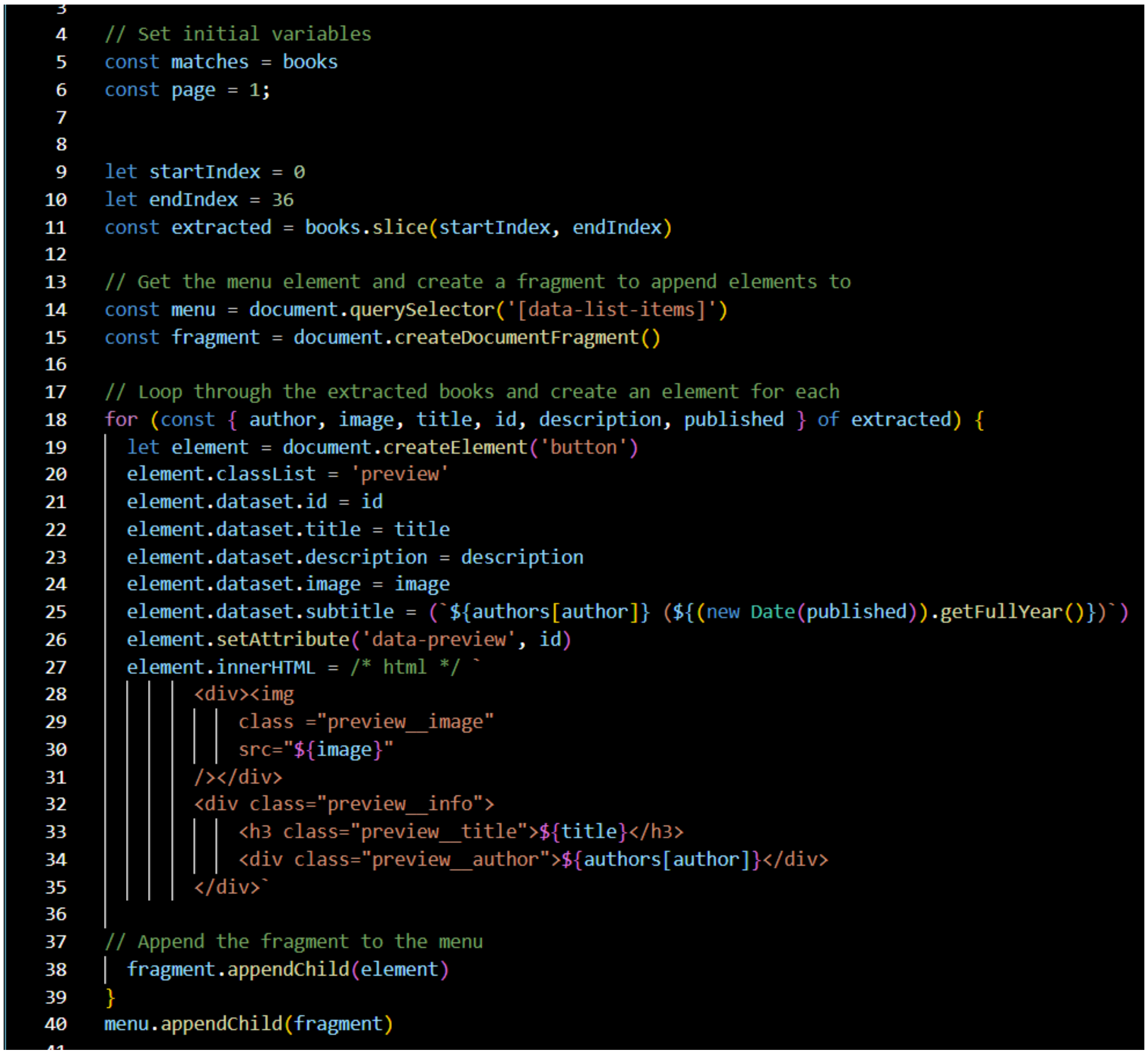
DWA_03.4 Knowledge Check_DWA3.1

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



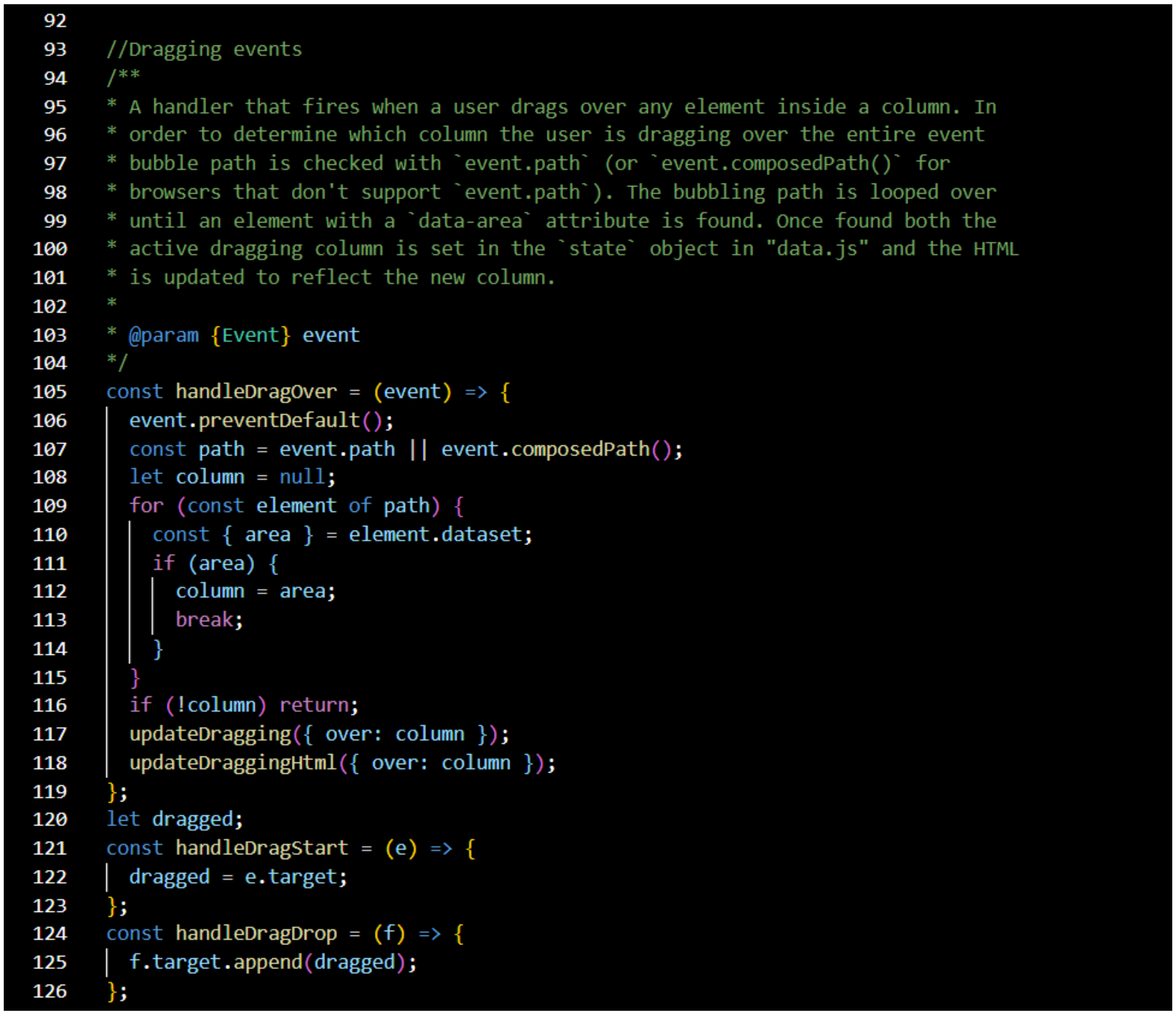
```markdown
# Fizz_Buzz

The program is designed to output a sequence of numbers from 1 to 100, but with a twist. Instead of printing the actual number, certain rules determine what should be printed. If a number is a multiple of 3, the program will print "Fizz" instead. Similarly, if a number is a multiple of 5, it will print "Buzz." However, if a number is a multiple of both 3 and 5, it will print "FizzBuzz." By following these rules, the program generates a sequence of outputs that replaces specific numbers with the corresponding words or phrases, resulting in a unique pattern of Fizz, Buzz, and FizzBuzz in place of the original numbers. This program is often used as a simple coding exercise to practice conditional statements and looping constructs in programming languages.

## Features

- 🔢Calculations
- 👍Minamal Fucntionality
- 💻Displays solution in Console dev

## Requirements

- An Intergrated Development Environment like Visual Studio Code.
- Basic [HTML,Javascript] skills.
- A Browser like [Microsoft_Edge]

## Getting Started

1. Clone the Repository

```
git clone https://github.com/TUMII01/sides.git
```

2. Open `index.html`.

3. Run a localhost server

4. Open Developer Tools

5. View Console
```

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



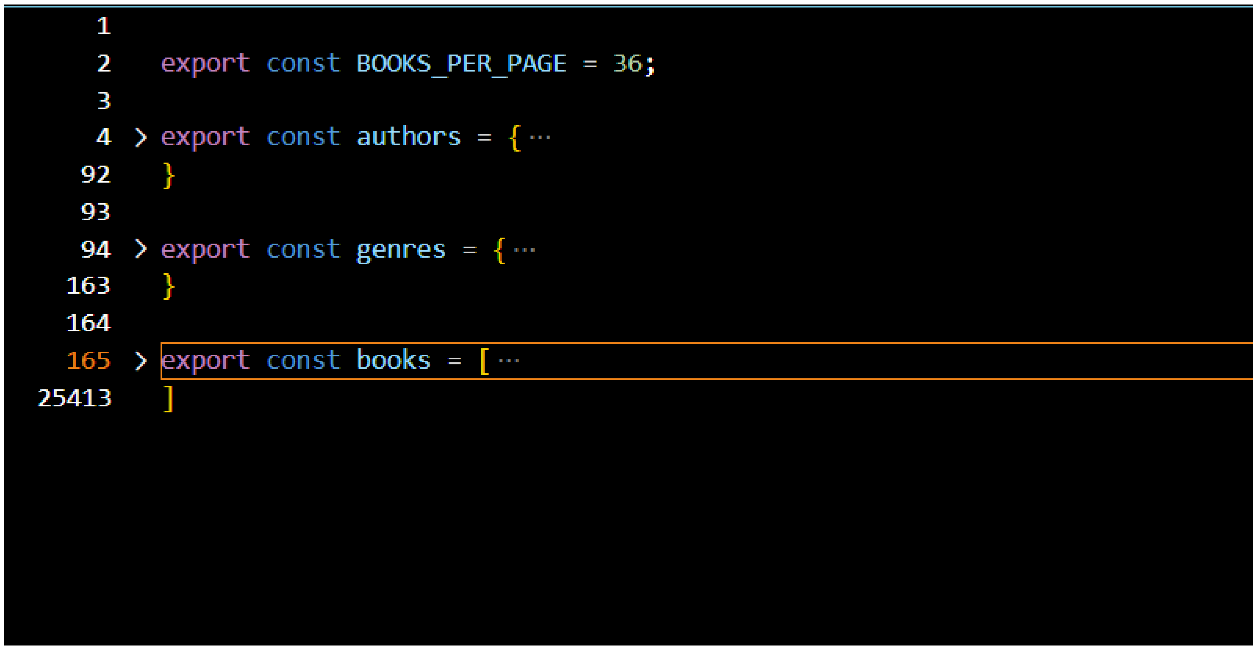
```
// Set initial variables
const matches = books
const page = 1;


let startIndex = 0
let endIndex = 36
const extracted = books.slice(startIndex, endIndex)

// Get the menu element and create a fragment to append elements to
const menu = document.querySelector('[data-list-items]')
const fragment = document.createDocumentFragment()

// Loop through the extracted books and create an element for each
for (const { author, image, title, id, description, published } of extracted) {
  let element = document.createElement('button')
  element.classList = 'preview'
  element.dataset.id = id
  element.dataset.title = title
  element.dataset.description = description
  element.dataset.image = image
  element.dataset.subtitle = (`${authors[author]} (${(new Date(published)).getFullYear()})`)
  element.setAttribute('data-preview', id)
  element.innerHTML = /* html */ `
        <div><img
            class ="preview__image"
            src="${image}"
        /></div>
        <div class="preview__info">
            <h3 class="preview__title">${title}</h3>
            <div class="preview__author">${authors[author]}</div>
        </div>`

// Append the fragment to the menu
  fragment.appendChild(element)
}
menu.appendChild(fragment)
```

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

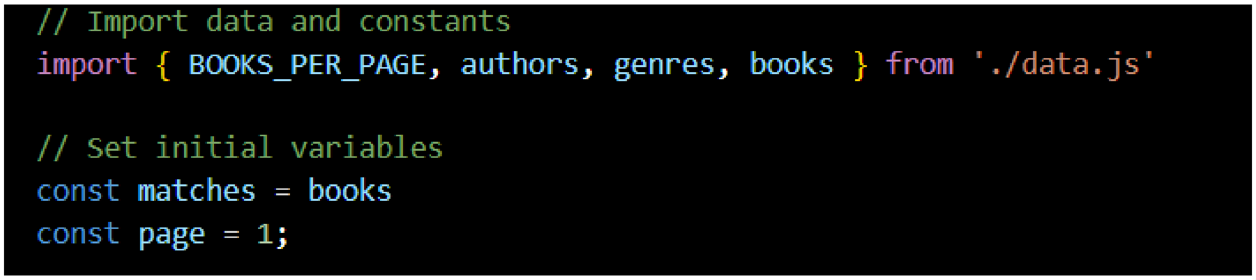


```

//Dragging events
/**
* A handler that fires when a user drags over any element inside a column. In
* order to determine which column the user is dragging over the entire event
* bubble path is checked with `event.path` (or `event.composedPath()` for
* browsers that don't support `event.path`). The bubbling path is looped over
* until an element with a `data-area` attribute is found. Once found both the
* active dragging column is set in the `state` object in "data.js" and the HTML
* is updated to reflect the new column.
*
* @param {Event} event
*/
const handleDragOver = (event) => {
  event.preventDefault();
  const path = event.path || event.composedPath();
  let column = null;
  for (const element of path) {
    const { area } = element.dataset;
    if (area) {
      column = area;
      break;
    }
  }
  if (!column) return;
  updateDragging({ over: column });
  updateDraggingHtml({ over: column });
};
let dragged;
const handleDragStart = (e) => {
  dragged = e.target;
};
const handleDragDrop = (f) => {
  f.target.append(dragged);
};
```

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



```
export const BOOKS_PER_PAGE = 36;

export const authors = {···
}

export const genres = {···
}

export const books = [···
]
```



```
// Import data and constants
import { BOOKS_PER_PAGE, authors, genres, books } from './data.js'

// Set initial variables
const matches = books
const page = 1;
```