




DWA_03.4 Knowledge Check_DWA3.1


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

 github.com/OnalennaRammapatsi/ONARAM970_BCL2302_GroupJustin_OnalennaRammapatsi_FinalCapStone

 README.md 

[FINAL CAPSTONE.pdf](#)

ONARAM970_BCL2302_GroupJustin_OnalennaRammapatsi_FinalCapStone

 Webapp to be audited as part of final assessment for first JS module

Sure, here's an example Readme file for a JavaScript code that displays a list of books and provides some features to interact with the list:

Book List Web Page

This is a web page that displays a list of books and provides some features to interact with the list. The web page is built using HTML, CSS, and JavaScript.

Technologies Used

- **HTML:** The web page's structure is built using HTML, including the use of semantic elements to structure the content.
- **CSS:** The web page's styling is done using CSS, including the use of CSS Grid and Flexbox to layout the content.
- **JavaScript:** The web page's interactivity is implemented using JavaScript, including the use of the Document Object Model (DOM) to manipulate the web page's content and respond to user actions.

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

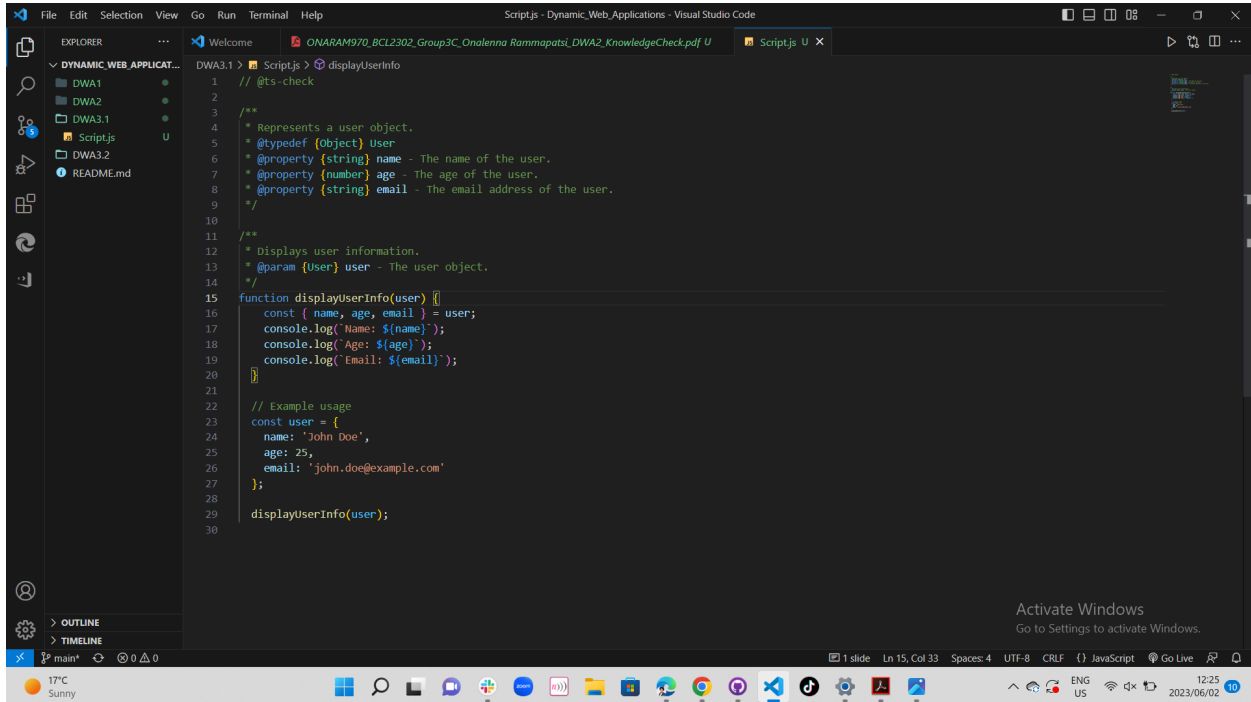
```
1 scripts.js > addEventListener("submit") callback
2 import { genres } from './data.js'
3 import { books } from './data.js'
4 const matches = books //declared my variables with their values
5 let page = 1;
6 if (!books && !Array.isArray(books)) { throw new Error("Source required") }
7 if (!page && page.length < 2) { throw new Error("Range must be an array with two numbers") }
8 //declared the objects day and night with their RGB values
9 const day = {
10   dark: '10, 10, 20',
11   light: '255, 255, 255',
12 }
13 const night = {
14   dark: '255, 255, 255',
15   light: '10, 10, 20',
16 }
17 const fragment = document.createDocumentFragment() //The new DocumentFragment is created and used to hold the newly created div
18 //that will contain the previes of the books*/
19
20 let startIndex = 0; //Initialized 2 variables 'startIndex' and 'endIndex' to 0 and 36.
21 let endIndex = 36;
22 const extracted = books.slice(startIndex, endIndex) // .slice returns elements in an array, as a new array.
23 for (let i = 0; i < extracted.length; i++) {
24   const preview = document.createElement('button')
25   preview.className = 'preview'
26   preview.dataset.id = books[i].id
27   preview.dataset.title = books[i].title
28   preview.dataset.image = books[i].image
29   preview.dataset.subtitle = `${authors[books[i].author]} (${(new Date(books[i].published)).getFullYear())}`
30   preview.dataset.description = books[i].description
31   preview.dataset.genre = books[i].genres
32   //creates a template to structure the html then append the preview to fragment
33   preview.innerHTML = `html`
34   <div>
35     
36   </div>
37   <div class="preview_info">
38     <dt class="preview_title">${books[i].title}<dt>
39     <dt class="preview_author">By ${authors[books[i].author]}<dt>
40   `
41 }
```

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

```
1 // @ts-check
2
3 /**
4  * Calculates the sum of two numbers.
5  * @param {number} a - The first number.
6  * @param {number} b - The second number.
7  * @returns {number} The sum of the two numbers.
8  */
9 function calculateSum(a, b) {
10   return a + b;
11 }
12
13 // Testing the function
14 const result = calculateSum(5, 10);
15 console.log(result);
16
```

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

Using the Destructuring assignment



```
1 // @ts-check
2
3 /**
4  * Represents a user object.
5  * @typedef {Object} User
6  * @property {string} name - The name of the user.
7  * @property {number} age - The age of the user.
8  * @property {string} email - The email address of the user.
9  */
10
11 /**
12  * Displays user information.
13  * @param {User} user - The user object.
14  */
15 function displayUserInfo(user) {
16     const { name, age, email } = user;
17     console.log(`Name: ${name}`);
18     console.log(`Age: ${age}`);
19     console.log(`Email: ${email}`);
20 }
21
22 // Example usage
23 const user = {
24     name: 'John Doe',
25     age: 25,
26     email: 'john.doe@example.com'
27 };
28
29 displayUserInfo(user);
30
```

The screenshot shows the Visual Studio Code interface with a file explorer on the left showing a project named 'DYNAMIC_WEB_APPLICATION'. The main editor displays a TypeScript file 'Script.js' with a function 'displayUserInfo' that uses destructuring assignment to extract 'name', 'age', and 'email' from a 'user' object. The status bar at the bottom indicates the file is 'main.ts' and the current line is 15, column 33. The system tray at the bottom shows the date as 2023/06/02 and the time as 12:25.