

DWA_03.4 Knowledge Check_DWA3.1

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

Below is a screenshot of the readme file for the BookConnect app that we did as a capstone challenge for the IWA course. It shows how the markdown file was applied for the project.



```
# Table of Contents
Table of Contents (up to date)
- [About the App](#about-the-app)
- [Instructions on how to use the Book Connect App:]
  (#instructions-on-how-to-use-the-book-connect-app)
- [Slideshow Link](#slideshow-link)
- [Languages Used:](#languages-used)

<!-- omit in toc -->
# Author
Xola Maci
Student Number: XOLMAC202

# About the App
This an app were users can search for a book, preview the book and load more books that
they
may be interested in. Users are ables to change the settings of the app based on their
preference, wether set it to "Dark" or "Light" mode.

# Instructions on how to use the Book Connect App:

1. Open app and you(user) will see the list of different books displayed in the page,
with cover pages.
2. User can press the load more, to access more books.
3. Click on the book to get a detailed description of the book, the author and the year
the book was published.
4. Press the search icon/button to filter books accorging to genre, author.
5. User can press the profile icon to change between themes (day or night).
```

01:11 0 0 0 Quokka Ln 3, Col 34 Spaces: 4 UTF-8 CRLF Markdown Port: 5501 Prettier

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

I used code from the IWA17 challenge to add JSDOC comments to the function called addCell, in the screenshot below it shows how this was achieved.

```
/**existing is for the existing html content.
 * ClassString creates a table with a CSS class called 'classString'
 * (&nbsp;${value}&nbsp;) places the value inside the table cell, makes space around the value.
 * @param {string} existing - This is for the existing HTML content
 * @param {string} classString - creates a table with a CSS class called 'classString'
 * @param {string} value - This is the value that will added to the cell table
 * @returns {result} - This returns results as a string
 */
const addCell = (existing, classString, value) => {
  const result =`${existing}
    <td class="${classString}">
      &nbsp;${value}&nbsp;
    </td>
  `;
  return result;
};
```

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

I used the code from IWA8 to check for errors in the code using @ts-check and used the @typedef to define the object.

```
1  // @ts-check
2  const leoName = 'Leo Musvairé'
3  const leoNumber = '2'
4  const leoStreet = 'Church St.'
5  const leoPostal = '3105'
6  const leoBalance = '-10'
7
8  const sarahName = 'Sarah'
9  const sarahSurname = 'Kleinhans'
10 const sarahBalance = '-4582.21000111'
11 const sarahNumber = '13'
12 const sarahStreet = 'William Close'
13 const sarahPostal = '0310'
14
15 // Only change below this line
16
17 /**
18  * An Object with personal details of a user
19  * @typedef {object}
20  */
21 const leo = {
22   name: `${leoName}`,
23   balance: leoBalance,
24   accessId: '47afb389-8014-4d0b-aff3-e40203d2107f',
25   age: 24,
26   'address': {
27     number: leoNumber,
```

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

Another form of documentation that I implemented on my final Capstone (IWA19) were inline comments and descriptive/ easy to understand variable names and function names. The screenshot below shows inline comments and variable names:

```
// all the extracted data types from the html file
const activeList = document.querySelector("[data-list-active]");
const dataList = document.querySelector("[data-list-items]");
const btnClose = document.querySelector("[data-list-close]");
const btnCancel = document.querySelector("[data-search-cancel]");
const bookDescription = document.querySelector("[data-list-description]");
const bookSubtitle = document.querySelector("[data-list-subtitle]");
const bookTittle = document.querySelector("[data-list-title]");
const blurryBookPic = document.querySelector("[data-list-blur]");
const bookPic = document.querySelector("[data-list-image]");
const themeOption = document.querySelector("[data-settings-theme]");
const formChange = document.querySelector("[data-settings-form]");
const formOverlay = document.querySelector("[data-settings-overlay]");
const searchOverlay = document.querySelector("[data-search-overlay]");
const btnSettings = document.querySelector("[data-header-settings]");
const listMessage = document.querySelector("[data-list-message]");
```
