

DWA_03.4 Knowledge Check_DWA3.1

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

```
IWA17 > markdown.md > # Calendar Website > ## User stories
1 # Calendar Website
2
3 This calendar website is designed to be simple and user friendly, with the aim of providing just a minimal functionality of highlighting the current date
  so that the user can identify what day it is. The website provides all the basic features you would expect from a calendar, like the month and year as the
  header of the calendar, rows indicating weeks, columns indicating the days, the inner blocks indicating the date number of the month and the current date
  being highlighted.
4
5 ## Features
6
7 - Simple and minimal look
8 - Ability to know the current date
9
10 ## Requirements
11
12 - An IDE like [Visual Studio Code](https://code.visualstudio.com/).
13 - Basic [HTML, CSS and Javascript](https://developer.mozilla.org/en-US/docs/Learn).
14 - A browser like [Brave](https://brave.com/).
15
16 ## Getting Started
17
18 Clone the repository
19 ```
20 git clone https://github.com/RONHEL235/RONHEL235_FT02306_GroupB_RoneeHelepi_IWA
21 ```
22
23
24 Run localhost server
25
26 Open `index.html`
27
28 ## User stories
29
30 - As a user, I want to see the current month and year as a heading of the calendar so that I can see the current month and year we are in.
31 - As a user, I want to see the number of weeks in row form so that I can identify what week we are currently in.
```

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

```
// Only edit below
/**
 * The function `createArray` generates an array of integers
  where each element value is equal to its index in the array.
 * It initializes an empty array named `result`.
```

```

    * It then iterates from 0 up to, but not including, the
    specified `length`, incrementing the index `i` by 1 in each
    iteration.
    * Inside the loop, it adds the current value of `i` to the
    `result` array using the `push` method.
    * Finally, it returns the populated `result` array.
    * @param {number} length - The desired length of the output
    array.
    * @returns {number[]} result - An array of integers from 0 up
    to `length` - 1.
    */
const createArray = (length) => {
    const result = []

    for (let i = 0; i < length; i = i + 1) {
        result.push(i)
    }
    return result
    // return a list that takes in the argument length, does
    some looping and gives it back
}

```

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

```
// @ts-check
```

```
file Edit Selection View Go Run Terminal Help interactive-web-apps
markdown.md U JS scripts.js 4. M X
IWA17 > JS scripts.js > createData > value
1 // @ts-check
2
3 const MONTHS = [
4   'January',
5   'February',
6   'March',
7   'April',
8   'May',
9   'June',
10  'July',
11  'August',
12  'September',
13  'October',
14  'November',
15  'December',
16 ];
17
18 const getDaysInMonth = (date) => new Date(date.getFullYear(), date.getMonth() + 1, 0).getDate();
19
20 // Only edit below
21 /**
22  * The function 'createArray' generates an array of integers where each element value is equal to its index in the array.
23  * It initializes an empty array named 'result'.
24  * It then iterates from 0 up to, but not including, the specified 'length', incrementing the index 'i' by 1 in each iteration.
25  * Inside the loop, it adds the current value of 'i' to the 'result' array using the 'push' method.
26  * Finally, it returns the populated 'result' array.
27  * @param {number} length - The desired length of the output array.
28  * @returns {number[]} result - An array of integers from 0 up to 'length' - 1.
29  */
```

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

```
* @typedef {string|number} Value - The value to be inserted
into the new cell.
* @param {Value} value
```

```
81
82 /**
83  * The function 'addCell' appends an HTML table cell ('<td>') to the provided existing HTML string.
84  * 1. It takes three arguments: the existing HTML string, a class string to be added to the new cell, and a value to be inserted into the cell.
85  * 2. Using a template literal, it appends a new '<td>' element with the provided class string and value to the existing HTML string.
86  * 3. The value within the cell is surrounded by non-breaking spaces ('&nbsp;') for spacing purposes.
87  * 4. The resulting concatenated HTML string is then returned.
88  * @param {string} existing - The existing HTML string.
89  * @param {string} classString - The CSS class to be applied to the new cell.
90  * @typedef {string|number} Value - The value to be inserted into the new cell.
91  * @param {Value} value
92  * @returns {string} result - The updated HTML string with the appended table cell.
93  */
94 const addCell = (existing, classString, value) => {
95   const result = /* html */ `
96     ${existing}
97     <td class="${classString}">
98       &nbsp;${value}&nbsp;
99     </td>
100   `;
101
102   return result;
103 }
104
```