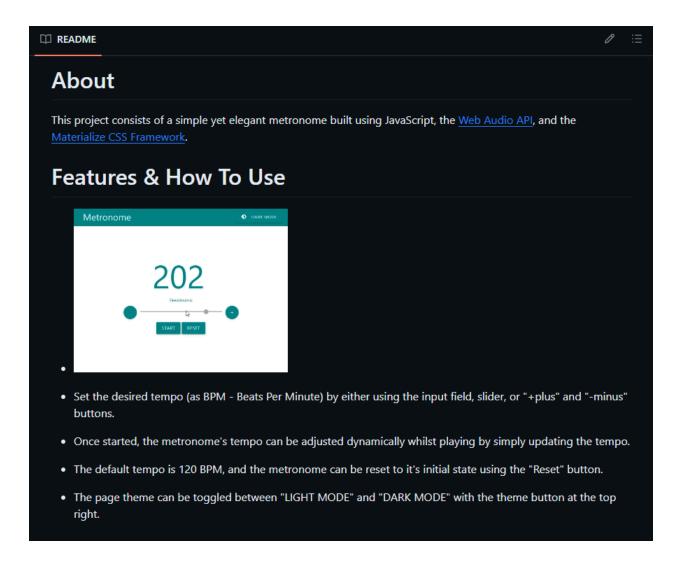
## DWA\_03.4 Knowledge Check\_DWA3.1

\_\_\_\_\_

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

```
🗓 README.md > 🖭 # How To Run
     # METRONOME
     # About
     This project consists of a simple yet elegant metronome built using JavaScript, the [Web Audio API]
     (https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API#web_audio_concepts_and_usage), and
     the [Materialize CSS Framework](https://materializecss.com/).
     # Features & How To Use
    - ![Alt text](assets/media/Metronome_Tut_1.gif)
     - Set the desired tempo (as BPM - Beats Per Minute) by either using the input field, slider, or "
     +plus" and "-minus" buttons.
     - Once started, the metronome's tempo can be adjusted dynamically whilst playing by simply updating
     the tempo.
     - The default tempo is 120 BPM, and the metronome can be reset to it's initial state using the
     "Reset" button.
     - The page theme can be toggled between "LIGHT MODE" and "DARK MODE" with the theme button at the
     top right.
     # Future features to be added
    - Volume control
    - Visualization of the tempo
     - Musical measures
    - Sound frequency control
     # How To Run
     -Clone the repository
     - Use a local server and browser to open the index.html or, if using VSCode, use the [Live Server
     Extension](https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer)
```



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

```
@property {Function} generatePreviewHtml - Function that creates and returns
* @param {Object} targetObject - Accepts any object literal that
* contains book information, in this case objects {@link book} from
* data.js, or once a search form has been submitted, {@link searchResult}
* @param {number} rangeStart - Defines where slicing of the targetObject
* should occur, which is calculated by taking the current {@link page} value
* and multiplying it by {@link BOOKS_PER_PAGE}
* @param {number} rangeEnd - Similar to rangeStart but defines where slicing
* @returns {DocumentFragment} listFragment - Returns the newly created document
* fragment which can then be appended to the list element.
export const generatePreviewHtml = (targetObject, rangeStart, rangeEnd) => {
  const listFragment = document.createDocumentFragment()
  const extractedBooks = targetObject.slice(rangeStart, rangeEnd)
  for (let { author, image, title, id } of extractedBooks) {
       author = authors[author] // to display the author as a name and not an id
      const preview = document.createElement('button')
       preview.className = 'preview
      preview.dataset.id = id
      preview.innerHTML = /* html */ `
               class="preview__image"
               src="${image}"
          <div class="preview__info">
              <div class="preview author">${author}</div>
       listFragment.appendChild(preview)
   return listFragment
```

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

In this snippet, the second and third parameters of "generatePreviewHtml" should be numbers. Using @ts-check, it shows an error because the second param is a string, and should be a number.

```
// @ts-check
import { generatePreviewHtml, html } from "./generators";

Argument of type 'string' is not assignable to parameter of type 'number'. ts(2345)

View Problem (Alt+F8) Quick Fix... (Ctrl+.)

generatePreviewHtml(html, "6", 7)
```

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

Added a Table of Contents to my READEME.md using markdown all in one extension:

```
# Book-Connect

Table of Contents (up to date)
- [About](#about)
- [User Stories](#user-stories)
- [Features](#features)
- [Requirements](#requirements)
- [How to use:](#how-to-use)
```