

DWA_03.4 Knowledge Check_DWA3.1

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

```
# Athlete Race Stats HTML Generator <!-- omit in toc -->

The Athlete Race Stats HTML Generator is a user-friendly tool that simplifies
the presentation of athlete information and their race statistics. It organizes
raw data into an easily readable format, highlighting detailed statistics for an
athlete's most recent race.

## Table of Contents <!-- omit in toc -->

- [Features](#features)
- [Getting Started](#getting-started)
  - [Prerequisites](#prerequisites)
  - [Usage](#usage)
- [Demo](#demo)

## Features

- 🔄 Converts raw athlete and race data into a structured format.
- 📊 Focuses on displaying detailed race statistics for recent races.
- 🖥️ Offers an intuitive and user-friendly interface for viewing athlete profiles and race stats.

## Getting Started

### Prerequisites

- An IDE like [Visual Studio Code](https://code.visualstudio.com).
- Basic [HTML, CSS and JavaScript](https://developer.mozilla.org/en-US/docs/Learn) skills.
- A web browser to view the athlete's profiles and recent stats like [Chrome](https://www.google.com/chrome).

### Usage

1. Clone the repository.

...
https://github.com/seancoder/SEANOK255_FTOW2306_GroupA_SeaNoko_IWA16.git
...

2. Open the `index.html` file in your web browser.
3. Browse and explore athlete profiles and their recent race statistics.

## Demo

![Screenshot of athlete profiles and recent race statistics on a web browser.](image.png)
```

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

```
/**
 * Converts an ISO date string to a date object, formats it to `d MMM YYYY`, and returns it as a string.
 *
 * @param {string} date An ISO date as a string.
 * @returns {string} A formatted date (`d MMM YYYY`) as a string.
 */
const toDateFormat = (date) => {
  const dateFormat = { dateStyle: "medium" };
  const toDate = new Date(date);
  return toDate.toLocaleDateString("en-GB", dateFormat);
};

/**
 * Takes in an array of laps in minutes and sums them together. The total minutes result is then
 * converted to a specific time format `hh:mm` and the time is returned as a string.
 *
 * @param {Array<number>} LapsInMinsArr An array of numbers representing minutes.
 * @returns {string} Returns the sum of passed numbers (minutes) as a time formatted string (hh:mm).
 */
const toTimeFormat = (lapsInMinsArr) => {
  let totalMins = undefined;

  lapsInMinsArr.forEach((lapMins) => {
    totalMins === undefined ? (totalMins = lapMins) : (totalMins += lapMins);
  });

  const hours = String(Math.floor(totalMins / 60));
  const mins = String(totalMins % 60);

  const formattedHours = hours.length === 1 ? hours.padStart(2, "0") : hours;
  const formattedMins = mins.length === 1 ? mins.padStart(2, "0") : mins;

  const time = `${formattedHours}:${formattedMins}`;
  return time;
};

/**
 * Represents a race as an object with information about the date and time of the race.
 *
 * @typedef {Object} Race
 * @property {string} date - The date of the race.
 * @property {Array<number>} time - An array of lap times in minutes achieved by the athlete during the race.
 */
```

```

/**
 * Represents an athlete's information and race statistics as an object.
 *
 * @typedef {Object} Athlete
 * @property {string} firstName - The athlete's first name.
 * @property {string} surname - The athlete's surname.
 * @property {string} id - A unique identifier assigned to the athlete.
 * @property {Array<Race>} races - An array of races (represented as {@link Race} objects) completed by the athlete.
 */

/**
 * Takes an object representing an athlete's information and race statistics. This function performs some calculations
 * to determine the total number of races and the details of the athlete's most recent race. It utilizes the
 * {@link toDateFormat} and {@link toTimeFormat} functions internally to format the date and time of the recent race.
 * The athlete's ID, first name, surname, total races, and recent race statistics (date and time) are then appended to
 * a document fragment, which is returned by the function.
 *
 * @param {Athlete} athlete - An object containing the athlete's information and race statistics as defined by the
 * {@link Athlete} type.
 * @returns {DocumentFragment} A document fragment to be appended to the HTML DOM.
 */
const createHtml = (athlete) => {
  const { firstName, surname, id, races } = athlete;
  const racesCopy = JSON.parse(JSON.stringify(races));
  const [{ date: recentRaceDate, time: recentRaceTime }] = racesCopy.reverse();

  const totalRaces = races.length;
  const date = toDateFormat(recentRaceDate);
  const time = toTimeFormat(recentRaceTime);

  const fragment = document.createDocumentFragment();

  const header = document.createElement("h2");
  const dList = document.createElement("dl");
  header.textContent = id;
  dList.innerHTML = `
<dt>Athlete:</dt>
<dd>${firstName} ${surname}</dd>

<dt>Total Races:</dt>
<dd>${totalRaces}</dd>

<dt>Event Date (Latest):</dt>
<dd>${date}</dd>

<dt>Total Time (Latest):</dt>
<dd>${time}</dd>
`;

  fragment.append(header, dList);
  return fragment;
};

```

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

```
// @ts-check

/**
 * Converts an ISO date string to a date object, formats it to `d MMM YYYY`, and returns it as a string.
 *
 * @param {string} date An ISO date as a string.
 * @returns {string} A formatted date (`d MMM YYYY`) as a string.
 */
const toDateFormat = (date) => {
  const dateFormat = { dateStyle: "medium" };
  const toDate = new Date(date);
  return toDate.toLocaleDateString("en-GB", dateFormat);
};

/**
 * Takes in an array of laps in minutes and sums them together. The total minutes result is then
 * converted to a specific time format `hh:mm` and the time is returned as a string.
 *
 * @param {Array<number>} LapsInMinsArr An array of numbers representing minutes.
 * @returns {string} Returns the sum of passed numbers (minutes) as a time formatted string (hh:mm).
 */
const toTimeFormat = (lapsInMinsArr) => {
  let totalMins = undefined;

  lapsInMinsArr.forEach((lapMins) => {
    totalMins === undefined ? (totalMins = lapMins) : (totalMins += lapMins);
  });
}
```