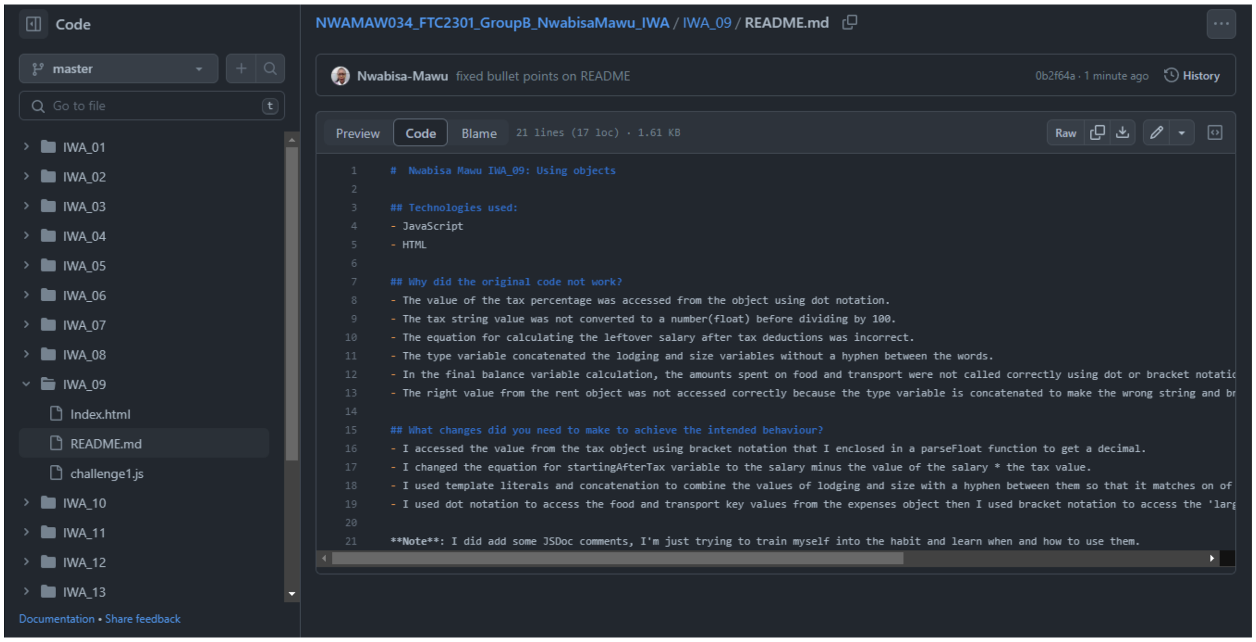
WA_03.4 Knowledge Check_DWA3.1

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

This is the code used to make the title headings and the questions.



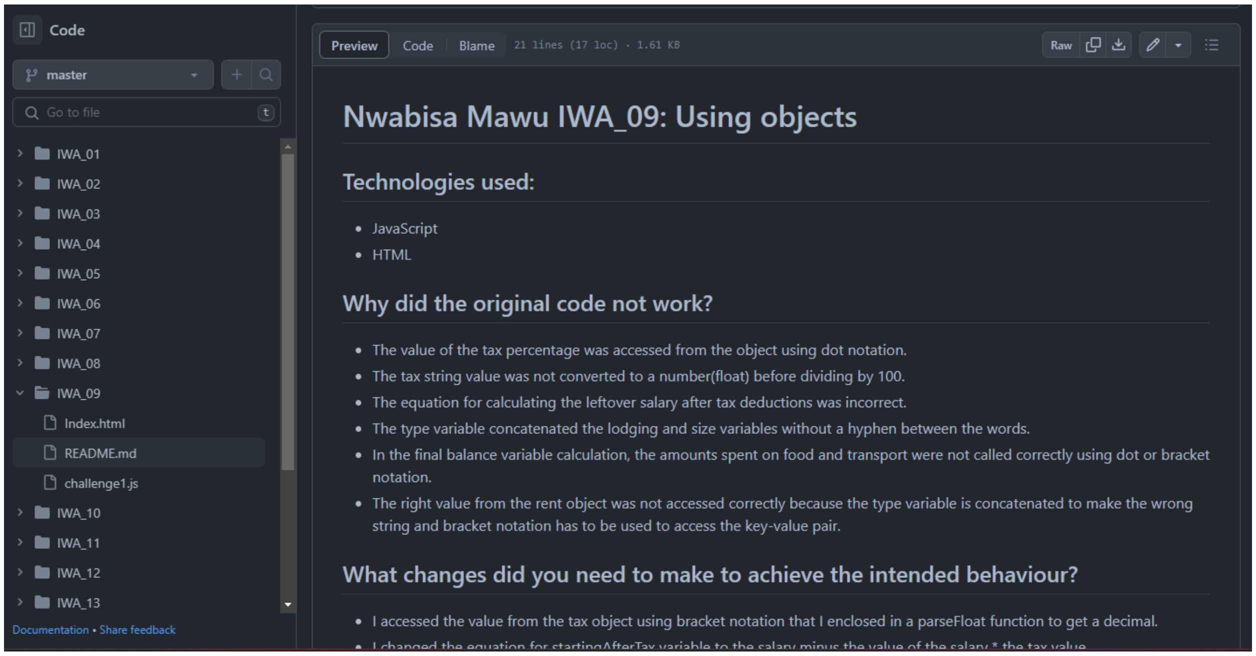
NWAMAW034_FTC2301_GroupB_NwabisaMawu_IWA / IWA_09 / README.md

Nwabisa-Mawu fixed bullet points on README 0b2f64a · 1 minute ago History

21 lines (17 loc) · 1.61 KB

```
#  Nwabisa Mawu IWA_09: Using objects

## Technologies used:
- JavaScript
- HTML

## Why did the original code not work?
- The value of the tax percentage was accessed from the object using dot notation.
- The tax string value was not converted to a number(float) before dividing by 100.
- The equation for calculating the leftover salary after tax deductions was incorrect.
- The type variable concatenated the lodging and size variables without a hyphen between the words.
- In the final balance variable calculation, the amounts spent on food and transport were not called correctly using dot or bracket notatic
- The right value from the rent object was not accessed correctly because the type variable is concatenated to make the wrong string and br

## What changes did you need to make to achieve the intended behaviour?
- I accessed the value from the tax object using bracket notation that I enclosed in a parseFloat function to get a decimal.
- I changed the equation for startingAfterTax variable to the salary minus the value of the salary * the tax value.
- I used template literals and concatenation to combine the values of lodging and size with a hyphen between them so that it matches on of
- I used dot notation to access the food and transport key values from the expenses object then I used bracket notation to access the 'larg

**Note**: I did add some JSDoc comments, I'm just trying to train myself into the habit and learn when and how to use them.
```

This is the preview of the readme file with the code above.



21 lines (17 loc) · 1.61 KB

Nwabisa Mawu IWA_09: Using objects

Technologies used:

- JavaScript
- HTML

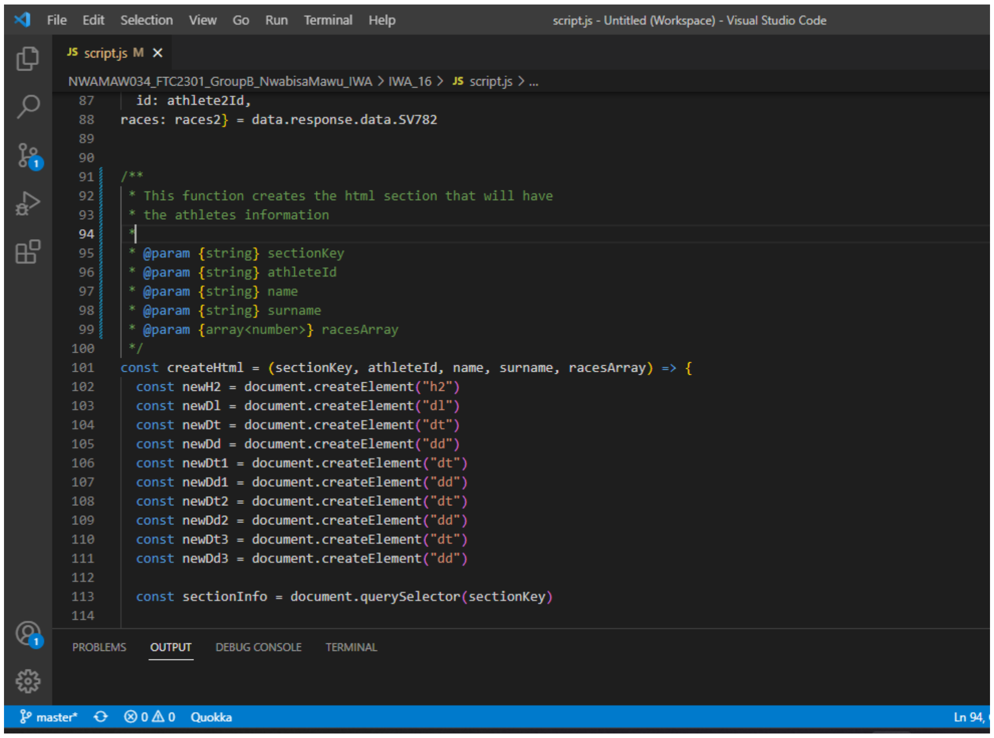
Why did the original code not work?

- The value of the tax percentage was accessed from the object using dot notation.
- The tax string value was not converted to a number(float) before dividing by 100.
- The equation for calculating the leftover salary after tax deductions was incorrect.
- The type variable concatenated the lodging and size variables without a hyphen between the words.
- In the final balance variable calculation, the amounts spent on food and transport were not called correctly using dot or bracket notation.
- The right value from the rent object was not accessed correctly because the type variable is concatenated to make the wrong string and bracket notation has to be used to access the key-value pair.

What changes did you need to make to achieve the intended behaviour?

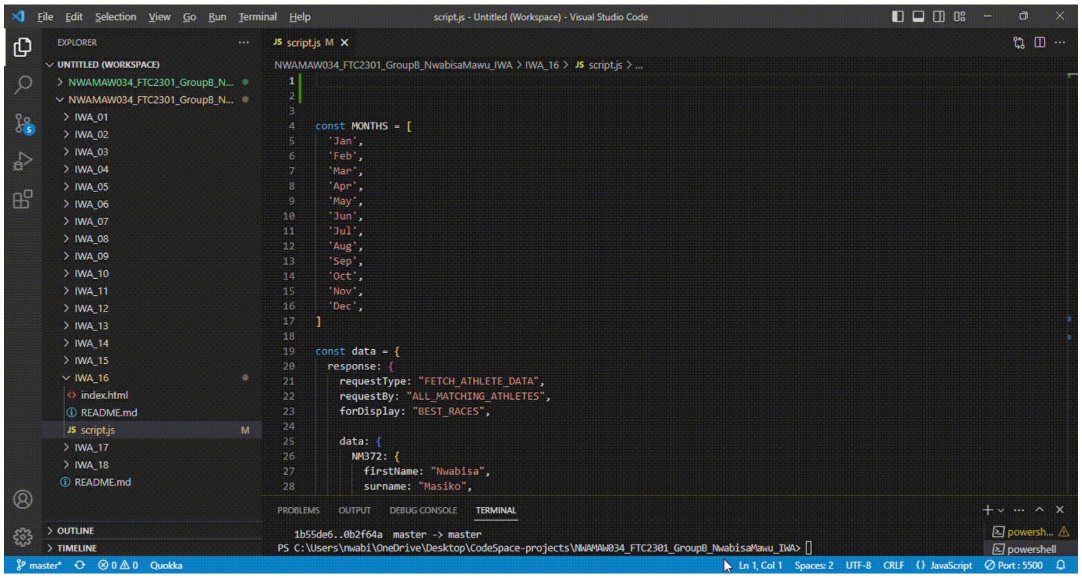
- I accessed the value from the tax object using bracket notation that I enclosed in a parseFloat function to get a decimal.
- I changed the equation for startingAfterTax variable to the salary minus the value of the salary * the tax value.

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



```
  id: athlete2Id,
races: races2} = data.response.data.SV782


/**
 * This function creates the html section that will have
 * the athletes information
 *
 * @param {string} sectionKey
 * @param {string} athleteId
 * @param {string} name
 * @param {string} surname
 * @param {array<number>} racesArray
 */
const createHtml = (sectionKey, athleteId, name, surname, racesArray) => {
  const newH2 = document.createElement("h2")
  const newDl = document.createElement("dl")
  const newDt = document.createElement("dt")
  const newDd = document.createElement("dd")
  const newDt1 = document.createElement("dt")
  const newDd1 = document.createElement("dd")
  const newDt2 = document.createElement("dt")
  const newDd2 = document.createElement("dd")
  const newDt3 = document.createElement("dt")
  const newDd3 = document.createElement("dd")

  const sectionInfo = document.querySelector(sectionKey)

```

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

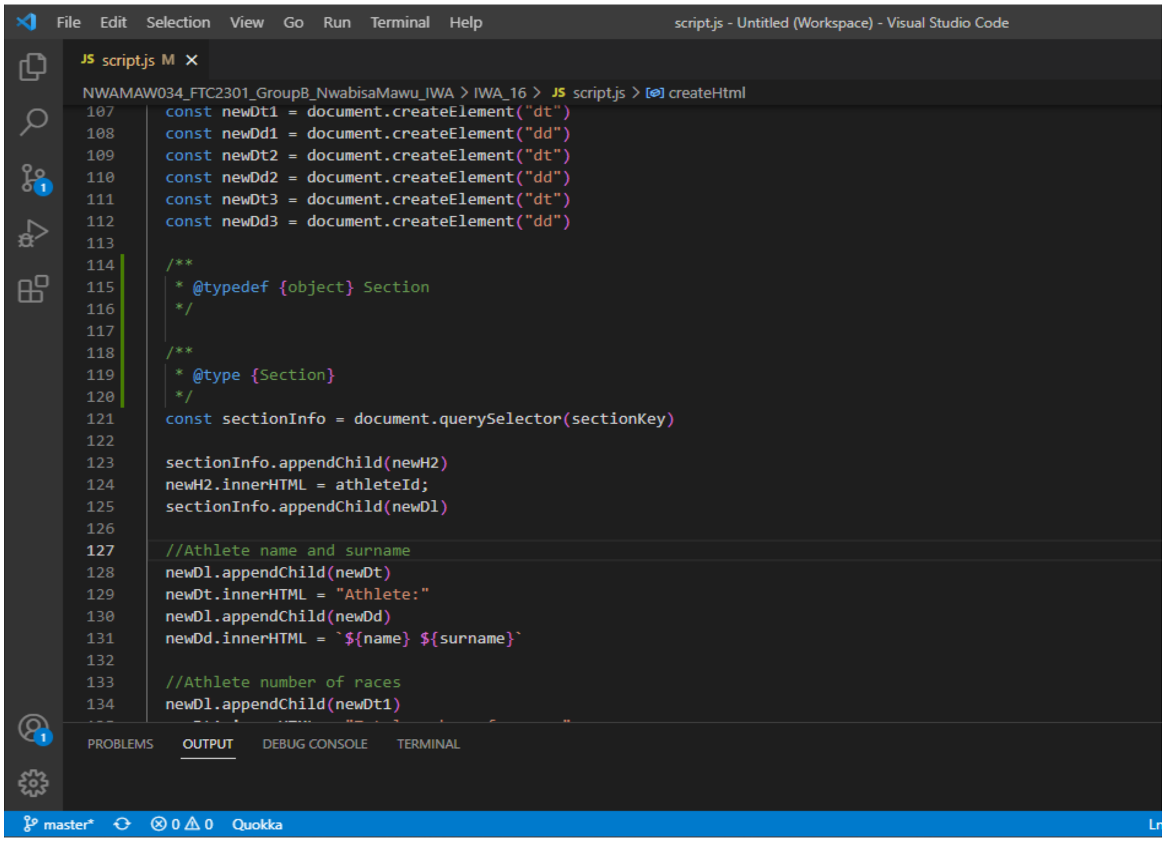


```



const MONTHS = [
  'Jan',
  'Feb',
  'Mar',
  'Apr',
  'May',
  'Jun',
  'Jul',
  'Aug',
  'Sep',
  'Oct',
  'Nov',
  'Dec',
]

const data = {
  response: {
    requestType: "FETCH_ATHLETE_DATA",
    requestBy: "ALL_MATCHING_ATHLETES",
    forDisplay: "BEST_RACES",

    data: {
      NM372: {
        firstName: "Nwabisa",
        surname: "Masiko",
```

```
   1b55de6..0b2f64a  master -> master
PS C:\Users\nwabi\OneDrive\Desktop\CodeSpace-projects\NWAMAW034_FTC2301_GroupB_NwabisaMawu_IWA>
```

The parameter types in the {} had a * before them so when the type check is activated with @ts-check, the type declarations inside the brackets were highlighted and had to be fixed.

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


File
Edit
Selection
View
Go
Run
Terminal
Help
script.js - Untitled (Workspace) - Visual Studio Code
JS script.js M
NWAMAW034_FTC2301_GroupB_NwabisaMawu_IWA > IWA_16 > JS script.js > createHtml
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
master*
⊗ 0 ⚠ 0
Quokka

```javascript
    const newDt1 = document.createElement("dt")
    const newDd1 = document.createElement("dd")
    const newDt2 = document.createElement("dt")
    const newDd2 = document.createElement("dd")
    const newDt3 = document.createElement("dt")
    const newDd3 = document.createElement("dd")

    /**
     * @typedef {object} Section
     */

    /**
     * @type {Section}
     */
    const sectionInfo = document.querySelector(sectionKey)

    sectionInfo.appendChild(newH2)
    newH2.innerHTML = athleteId;
    sectionInfo.appendChild(newDl)

    //Athlete name and surname
    newDl.appendChild(newDt)
    newDt.innerHTML = "Athlete:"
    newDl.appendChild(newDd)
    newDd.innerHTML = `${name} ${surname}`

    //Athlete number of races
    newDl.appendChild(newDt1)
```