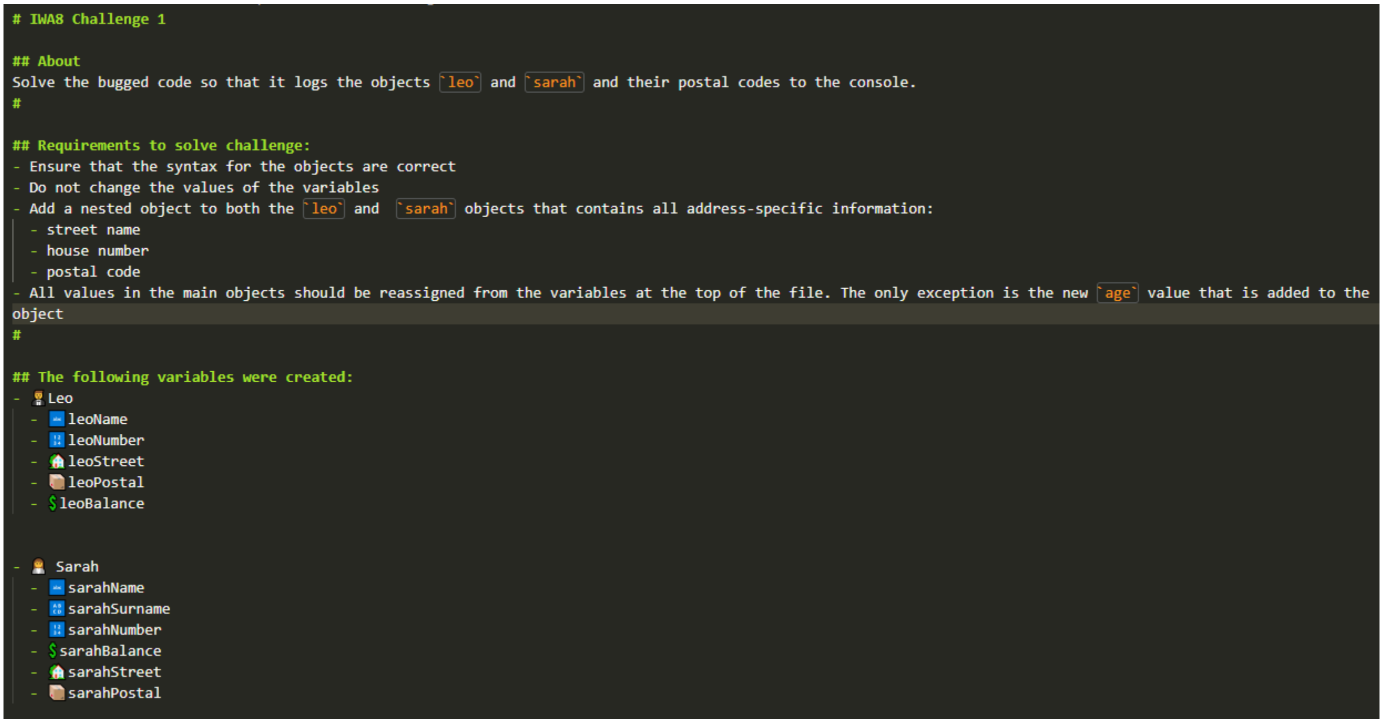
DWA_03.4 Knowledge Check_DWA3.1

---

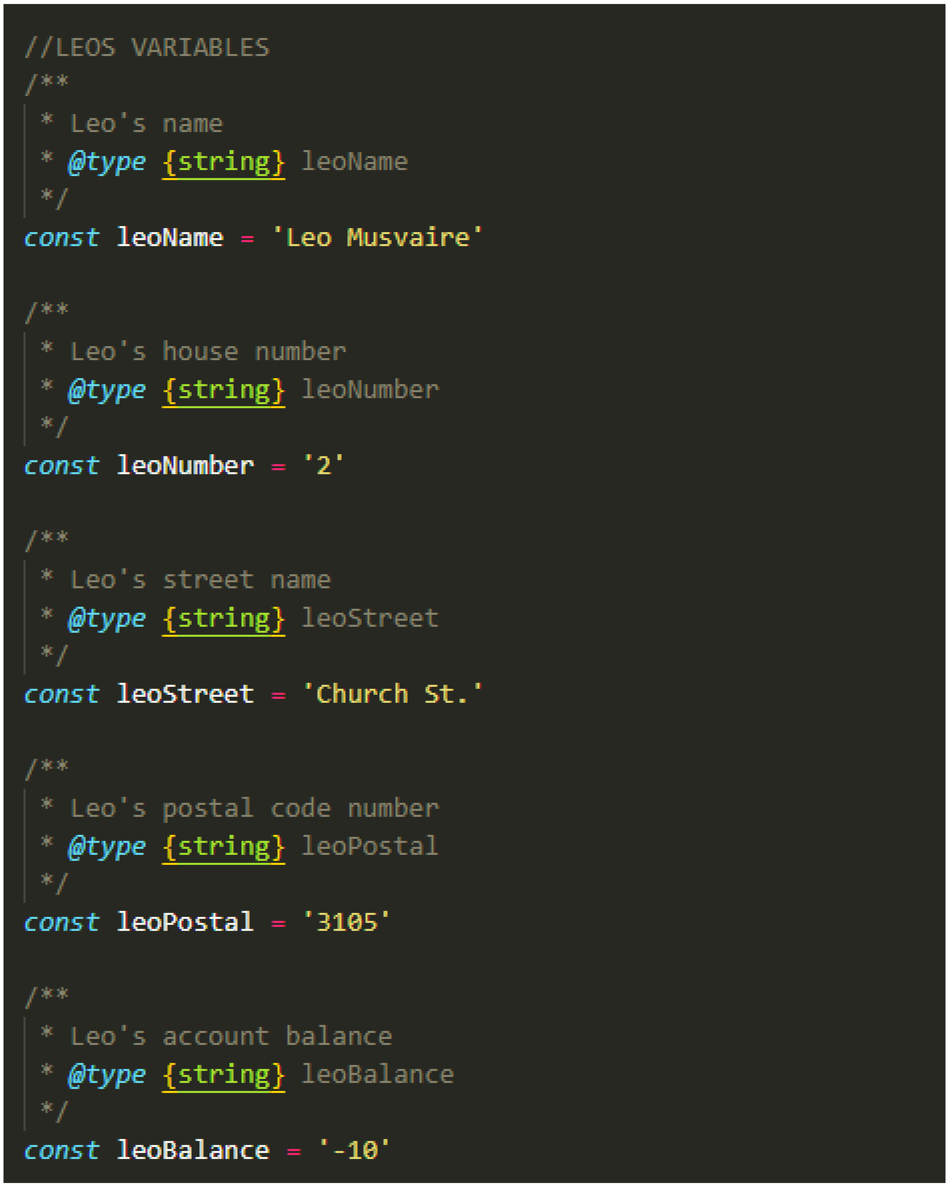
1. Please show how you applied a Markdown File to a piece of your code.



```
# IWA8 Challenge 1

## About
Solve the bugged code so that it logs the objects `leo` and `sarah` and their postal codes to the console.
#

## Requirements to solve challenge:
- Ensure that the syntax for the objects are correct
- Do not change the values of the variables
- Add a nested object to both the `leo` and  `sarah` objects that contains all address-specific information:
  - street name
  - house number
  - postal code
- All values in the main objects should be reassigned from the variables at the top of the file. The only exception is the new `age` value that is added to the object
#

## The following variables were created:
- 👨‍💼Leo
  - 🔡leoName
  - 🔢leoNumber
  - 🏡leoStreet
  - 📦leoPostal
  - 💲leoBalance


- 👩‍💼 Sarah
  - 🔡sarahName
  - 🔠sarahSurname
  - 🔢sarahNumber
  - 💲sarahBalance
  - 🏡sarahStreet
  - 📦sarahPostal
```

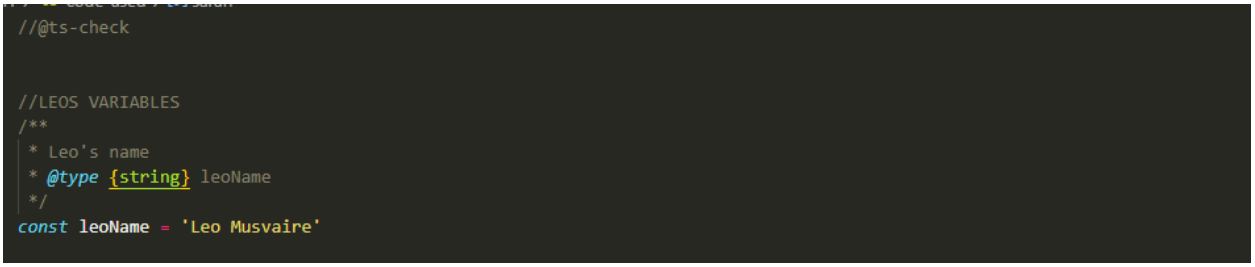
---

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



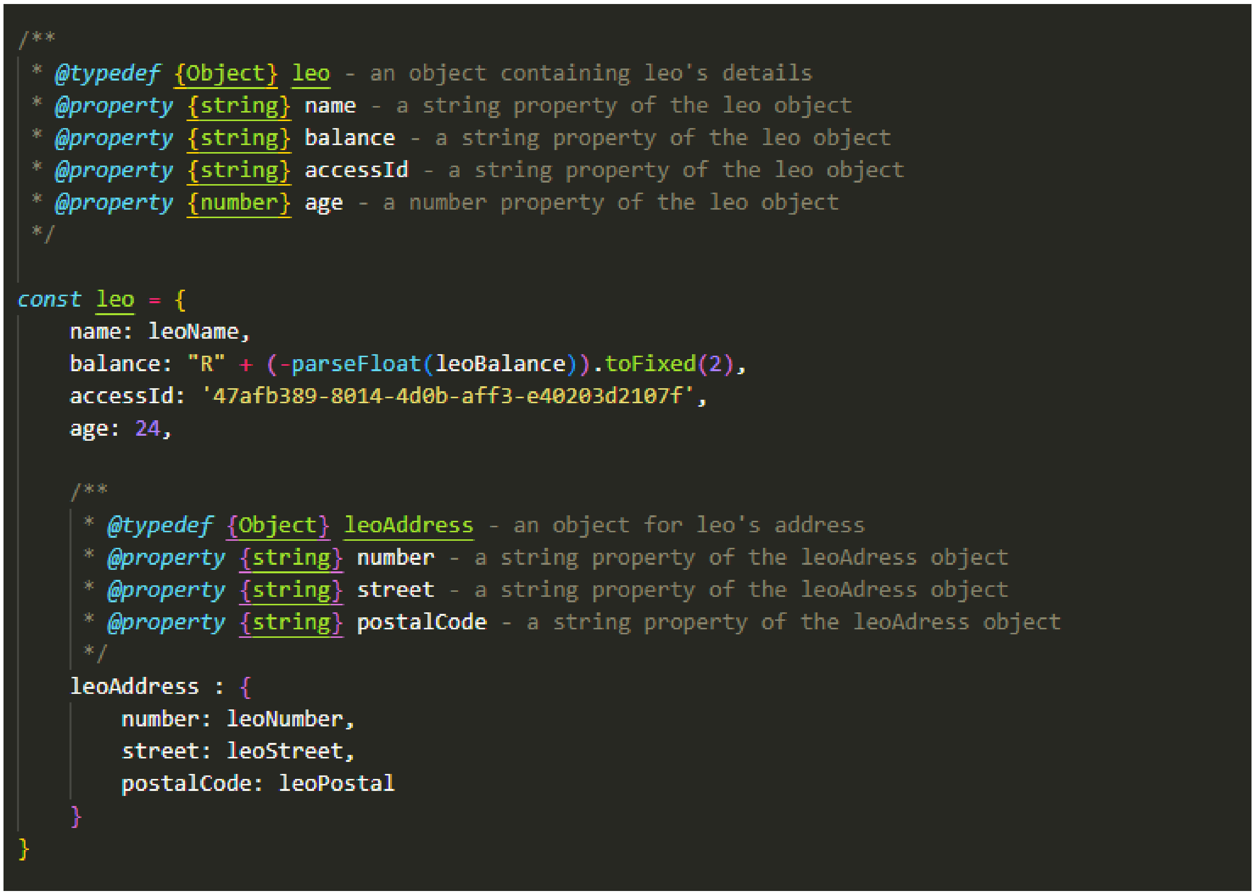
```
//LEOS VARIABLES
/**
 * Leo's name
 * @type {string} leoName
 */
const leoName = 'Leo Musvaire'

/**
 * Leo's house number
 * @type {string} leoNumber
 */
const leoNumber = '2'

/**
 * Leo's street name
 * @type {string} leoStreet
 */
const leoStreet = 'Church St.'

/**
 * Leo's postal code number
 * @type {string} leoPostal
 */
const leoPostal = '3105'

/**
 * Leo's account balance
 * @type {string} leoBalance
 */
const leoBalance = '-10'
```

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



```
//@ts-check


//LEOS VARIABLES
/**
 * Leo's name
 * @type {string} leoName
 */
const leoName = 'Leo Musvaire'
```

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



```
/**
 * @typedef {Object} leo - an object containing leo's details
 * @property {string} name - a string property of the leo object
 * @property {string} balance - a string property of the leo object
 * @property {string} accessId - a string property of the leo object
 * @property {number} age - a number property of the leo object
 */

const leo = {
    name: leoName,
    balance: "R" + (-parseFloat(leoBalance)).toFixed(2),
    accessId: '47afb389-8014-4d0b-aff3-e40203d2107f',
    age: 24,

    /**
     * @typedef {Object} leoAddress - an object for leo's address
     * @property {string} number - a string property of the leoAdress object
     * @property {string} street - a string property of the leoAdress object
     * @property {string} postalCode - a string property of the leoAdress object
     */
    leoAddress : {
        number: leoNumber,
        street: leoStreet,
        postalCode: leoPostal
    }
}
```