



Tebogo Motlhale

SLIDES

Why did the code not run?



1. The `matches` variable is assigned to `books`, but `books` is not defined. This will throw a reference error.
2. The `page` variable is not defined. It should be initialized with a value.
3. In the first `if` statement, the condition should be checking if `!books` OR `!Array.isArray(books)`, not `&&`. This is because if `books` is not defined, `!books` will be true, and the code will throw an error before checking `!Array.isArray(books)`. Using `&&` will cause an error when trying to access `length` on an undefined variable.
4. The `range` variable is not defined, but it is used in the second `if` statement to check its length. This will throw a reference error.
5. The `day` and `night` variables are defined using an object literal syntax, but the properties are not properly defined. The correct syntax should be `property: value`.
6. The `for` loop in the `extracted` variable declaration is missing a variable declaration. It should be `for (let { author, image, title, id } of extracted)`.
7. In the `for` loop for creating the `genres` options, the loop is iterating over `Object.entries(genres)` instead of `genres`. This will cause an error when trying to access properties of `genres`.
8. In the same `for` loop, `value` and `text` are not defined before being used to assign to `element.value` and `element.innerText`, respectively. It is unclear what the intended values are supposed to be.
9. In the `authors` loop, `id` should be incremented, not `i`.
10. In the `data-settings-theme` assignment, the comparison operator `===` is used instead of the assignment operator `=`. This will not set the value of `data-settings-theme`.
11. In the same line, the ternary operator should use `:` instead of `|` to separate the two possible values.
12. In the `createPreview` call inside the `data-list-button` click event, `page x BOOKS_PER_PAGE` and `{page + 1} x BOOKS_PER_PAGE` are not valid syntax. They should be multiplied with `*`, not `x`.
13. In the `data-list-button` click event, the `matches` array is not sliced correctly. It should be sliced from `page * BOOKS_PER_PAGE` to `(page + 1) * BOOKS_PER_PAGE`.
14. In the `data-search-form` click event, the `filters` parameter should be removed from the function definition as it is not used. Additionally, the `for` loop for iterating over `booksList` is missing a variable declaration (`let`). The conditional statements for matching the filters are also not properly structured and will throw syntax errors. It is unclear what the intended logic is.

Change i made to make the code run



First, the matches variable is assigned to books, but books is not defined, which will throw a reference error. Additionally, the page variable is not defined and should be initialized with a value. In the first if statement, the condition should check if !books OR !Array.isArray(books) instead of &&. This is because using && will cause an error when trying to access length on an undefined variable. The range variable is not defined but is used in the second if statement to check its length. The day and night variables are defined using an object literal syntax, but the properties are not properly defined. The for loop in the extracted variable declaration is missing a variable declaration, and the loop for creating the genres options is iterating over Object.entries(genres) instead of genres. In the same loop, value and text are not defined before being used to assign to element.value and element.innerText, respectively. The authors loop should increment id instead of i. The data-settings-theme assignment should use the assignment operator = instead of the comparison operator ==. The createPreview call inside the data-list-button click event should use * instead of x to multiply values. The matches array is not sliced correctly in the same event and should be sliced from page * BOOKS_PER_PAGE to (page + 1) * BOOKS_PER_PAGE. In the data-search-form click event, the filters parameter should be removed from the function definition, and the conditional statements for matching the filters should be properly structured. Lastly, in the data-settings-overlay submit event, there is an extra) in data-settings-overlay), and the condition for setting data-settings-overlay.open should use = instead of ==.

Explanation of the code



The program starts by importing three objects, `authors`, `genres`, and `books` from the `data.js` file using ES6 destructuring syntax. The program then initializes a constant variable `matches` to equal the `books` array, and a variable `page` to equal `1`.

After this, the program checks if `books` is undefined or not an array, throwing an error if either of these conditions is true. Similarly, it checks if `page` is undefined or shorter than two characters and throws an error if this is the case.

Next, the program creates two objects, `day` and `night`, that store the RGB values for different color schemes. The program then creates a document fragment and sets the values of `startIndex` and `endIndex` to `0` and `36`, respectively.

The program then creates an extracted array using `startIndex` and `endIndex` to extract a portion of the `books` array. A `for` loop then iterates through this extracted array and creates a preview of each book using the `document.createElement` function. The HTML structure for each preview is stored in the `preview` constant variable as a template literal, which is then added to the document fragment using the `appendChild` method.



After this, the program listens for click events on two buttons, a search button and a settings button. When clicked, the search button sets the `display` style property of the search overlay to "block", and the settings button sets the `display` style property of the settings overlay to "block". Similarly, when the cancel button is clicked on either overlay, the `display` style property of the overlay is set to "none".

Next, the program creates a theme toggle using the `dataSettingsTheme` variable and listens for a click event on the save button. When clicked, the program checks the value of `dataSettingsTheme` and sets the background and text colors of the document accordingly.

The program then creates two dropdown menus for filtering by author and genre using the `authorSelect` and `genreSelect` variables, respectively.

Finally, the program creates a function `detailsToggle` that is used to display the details of individual books. When a book preview is clicked, the function sets the `display` style property of the details overlay to "block" and sets the innerHTML of various elements to the relevant book details. The program then listens for click events on the close button of the details overlay, setting the `display` style property of the overlay to "none" when clicked.