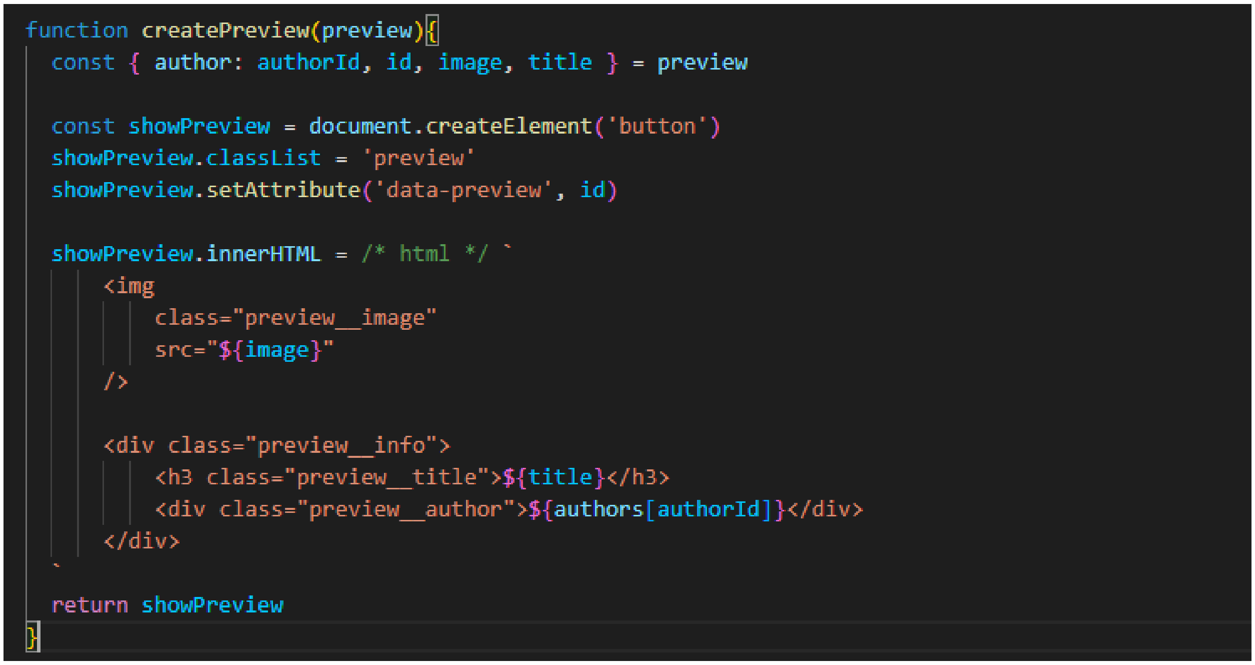
DWA_07.4 Knowledge Check_DWA7

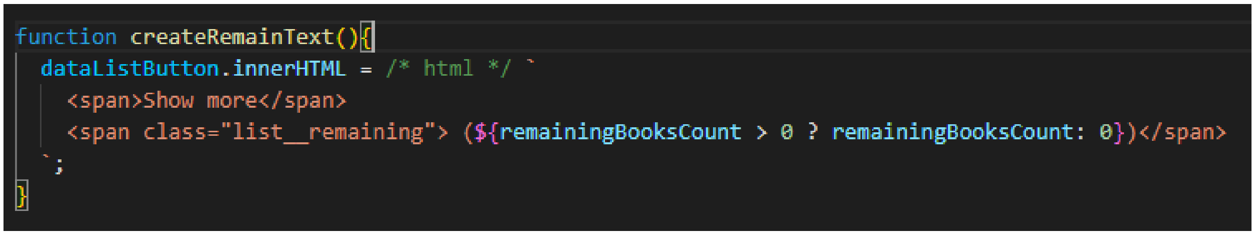
1. Which were the three best abstractions, and why?

- createPreview function helps minimize the code and duplication, it also makes it easier to change the code because all of the code is in one function in one place



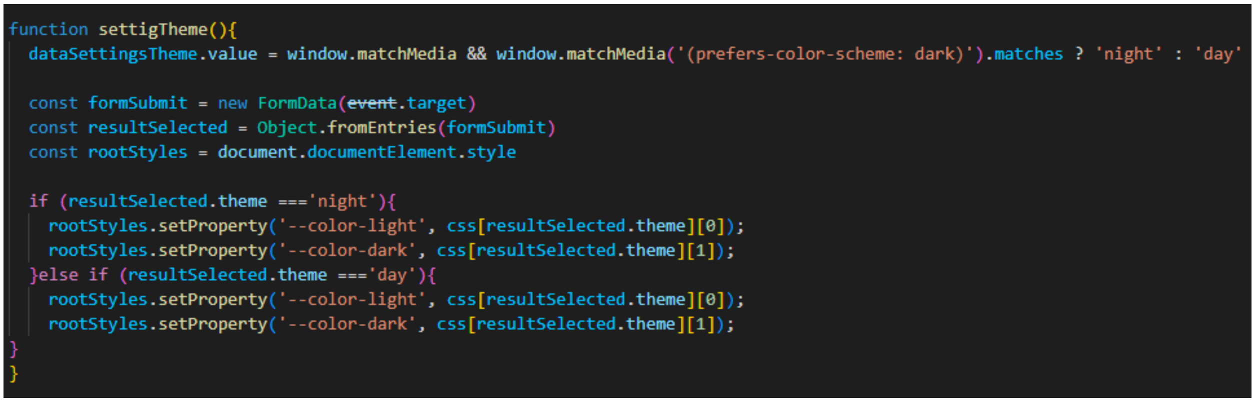
```
function createPreview(preview){
  const { author: authorId, id, image, title } = preview

  const showPreview = document.createElement('button')
  showPreview.classList = 'preview'
  showPreview.setAttribute('data-preview', id)

  showPreview.innerHTML = /* html */ `
      <img
          class="preview__image"
          src="${image}"
      />

      <div class="preview__info">
          <h3 class="preview__title">${title}</h3>
          <div class="preview__author">${authors[authorId]}</div>
      </div>
  `

  return showPreview
}
```

- Creating the remainText function saves a lot of time when updating the show more button because all of the code is only written once and be changed in one place if it is needed



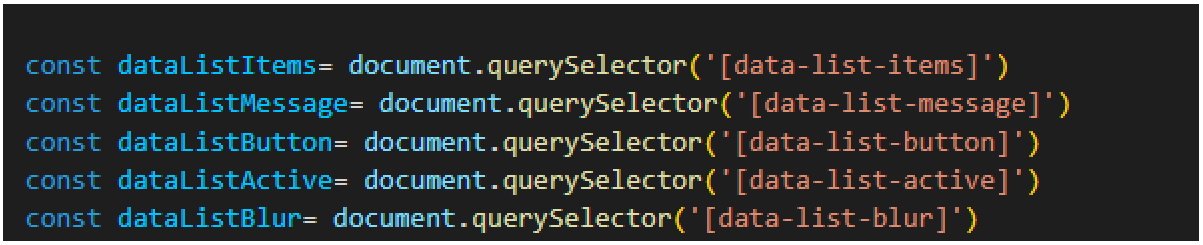
```
function createRemainText(){
  dataListButton.innerHTML = /* html */ `
    <span>Show more</span>
    <span class="list__remaining"> (${remainingBooksCount > 0 ? remainingBooksCount: 0})</span>
  `;
}
```

2. Which were the three worst abstractions, and why?



```
function settigTheme(){
  dataSettingsTheme.value = window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' : 'day'

  const formSubmit = new FormData(event.target)
  const resultSelected = Object.fromEntries(formSubmit)
  const rootStyles = document.documentElement.style

  if (resultSelected.theme ==='night'){
    rootStyles.setProperty('--color-light', css[resultSelected.theme][0]);
    rootStyles.setProperty('--color-dark', css[resultSelected.theme][1]);
  }else if (resultSelected.theme ==='day'){
    rootStyles.setProperty('--color-light', css[resultSelected.theme][0]);
    rootStyles.setProperty('--color-dark', css[resultSelected.theme][1]);
}
}
```

Abstraction in this case is helpful but not entirely necessary because the code is simple enough and is only used in one place and there is no need for duplication



```
const dataListItems= document.querySelector('[data-list-items]')
const dataListMessage= document.querySelector('[data-list-message]')
const dataListButton= document.querySelector('[data-list-button]')
const dataListActive= document.querySelector('[data-list-active]')
const dataListBlur= document.querySelector('[data-list-blur]')
```

Assigned the DOM element to a variable and made the code less to type out and easier to read

---

3. How can The three worst abstractions be improved via SOLID principles.

- split the function into smaller functions that can make the code more readable
- retrieve the DOM elements with a different function/ class and assign the values to variables

---