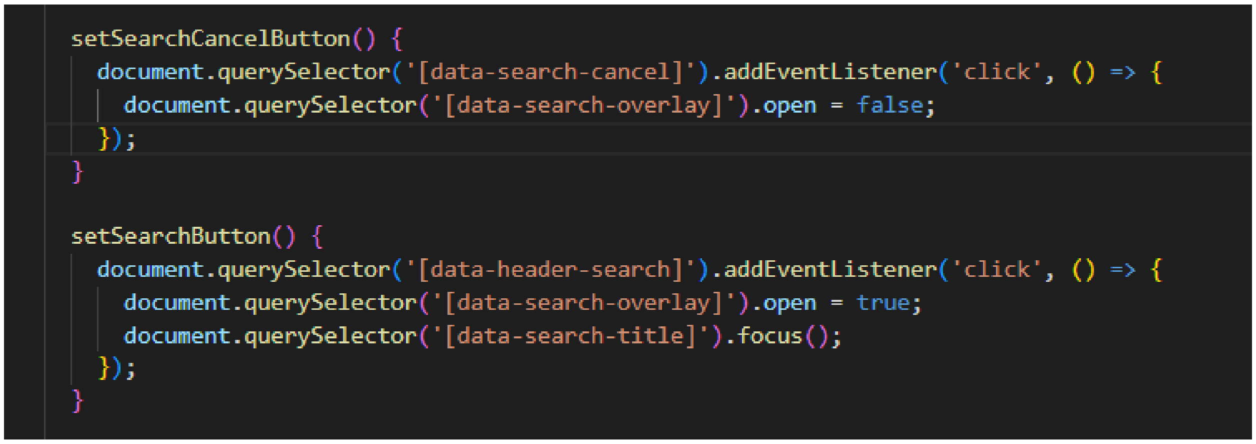
TB Mothibi_DWA7

DWA_07.4 Knowledge Check_DWA7

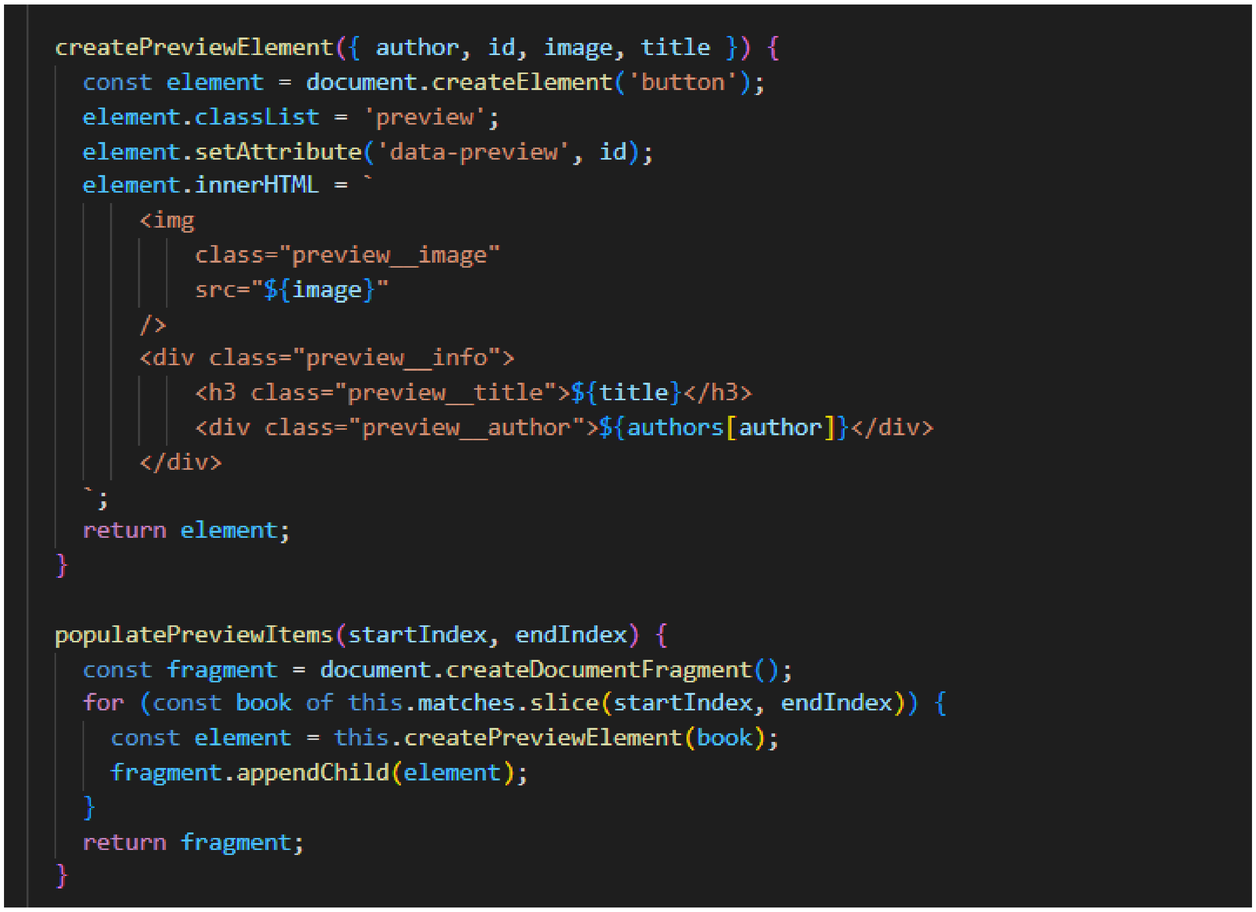
---

1. Which were the three best abstractions, and why?
    1. The functions/methods for overlays and cancel buttons



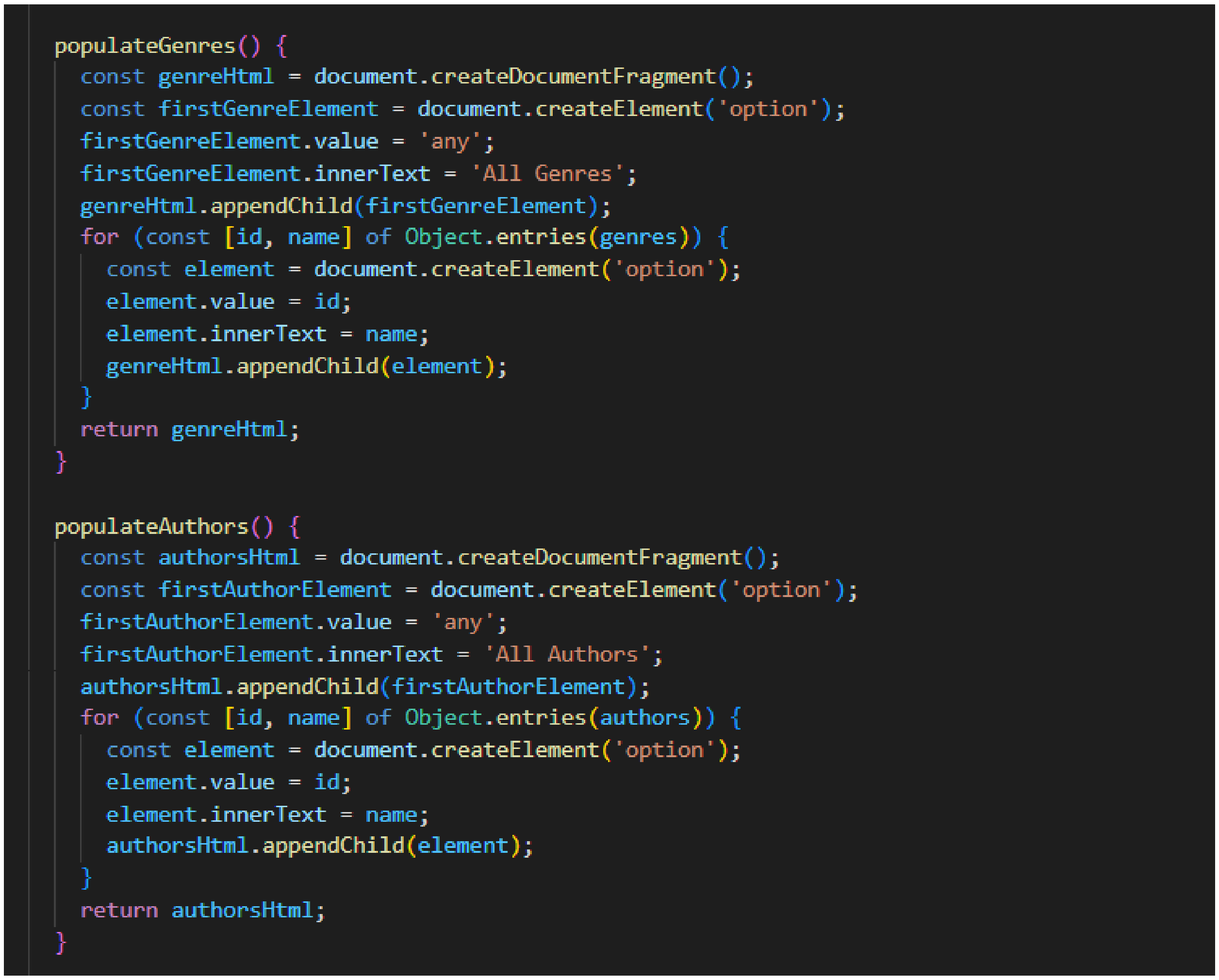
```
setSearchCancelButton() {
  document.querySelector('[data-search-cancel]').addEventListener('click', () => {
    document.querySelector('[data-search-overlay]').open = false;
  });
}

setSearchButton() {
  document.querySelector('[data-header-search]').addEventListener('click', () => {
    document.querySelector('[data-search-overlay]').open = true;
    document.querySelector('[data-search-title]').focus();
  });
}
```

2. It wasn't that complecated



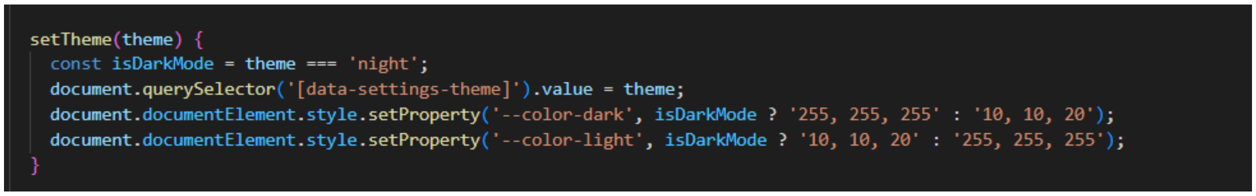
```
createPreviewElement({ author, id, image, title }) {
  const element = document.createElement('button');
  element.classList = 'preview';
  element.setAttribute('data-preview', id);
  element.innerHTML = `
      <img
          class="preview__image"
          src="${image}"
      />
      <div class="preview__info">
          <h3 class="preview__title">${title}</h3>
          <div class="preview__author">${authors[author]}</div>
      </div>
  `;
  return element;
}

populatePreviewItems(startIndex, endIndex) {
  const fragment = document.createDocumentFragment();
  for (const book of this.matches.slice(startIndex, endIndex)) {
    const element = this.createPreviewElement(book);
    fragment.appendChild(element);
  }
  return fragment;
}
```

2. Which were the three worst abstractions, and why?
   1. I wanted to make a function that holds just createElement and for loop, but it didn't work



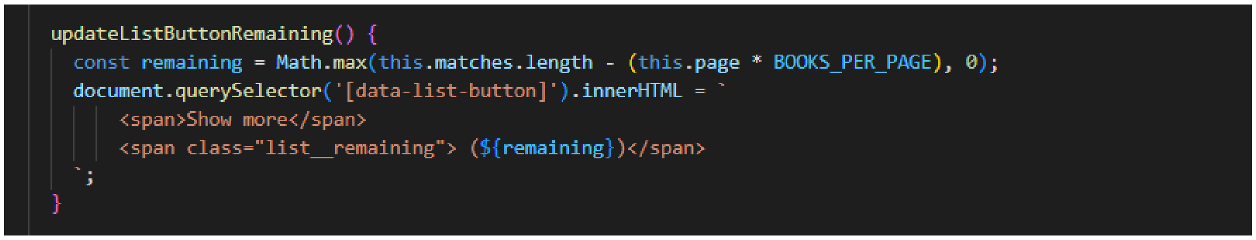
```
populateGenres() {
  const genreHtml = document.createDocumentFragment();
  const firstGenreElement = document.createElement('option');
  firstGenreElement.value = 'any';
  firstGenreElement.innerText = 'All Genres';
  genreHtml.appendChild(firstGenreElement);
  for (const [id, name] of Object.entries(genres)) {
    const element = document.createElement('option');
    element.value = id;
    element.innerText = name;
    genreHtml.appendChild(element);
  }
  return genreHtml;
}

populateAuthors() {
  const authorsHtml = document.createDocumentFragment();
  const firstAuthorElement = document.createElement('option');
  firstAuthorElement.value = 'any';
  firstAuthorElement.innerText = 'All Authors';
  authorsHtml.appendChild(firstAuthorElement);
  for (const [id, name] of Object.entries(authors)) {
    const element = document.createElement('option');
    element.value = id;
    element.innerText = name;
    authorsHtml.appendChild(element);
  }
  return authorsHtml;
}
```

2 Theme settings



```
setTheme(theme) {
  const isDarkMode = theme === 'night';
  document.querySelector('[data-settings-theme]').value = theme;
  document.documentElement.style.setProperty('--color-dark', isDarkMode ? '255, 255, 255' : '10, 10, 20');
  document.documentElement.style.setProperty('--color-light', isDarkMode ? '10, 10, 20' : '255, 255, 255');
}
```

3.

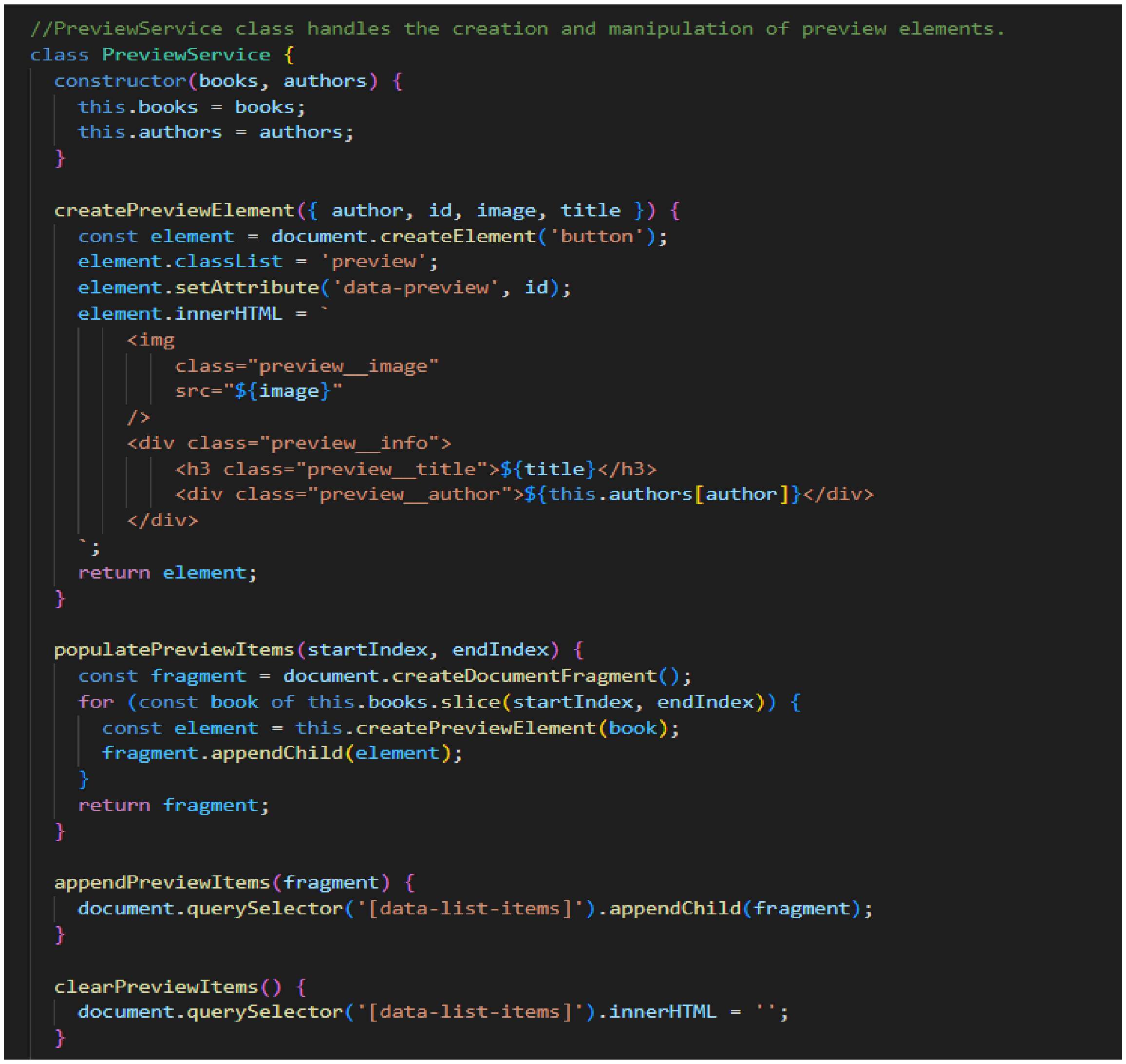


```
updateListButtonRemaining() {
  const remaining = Math.max(this.matches.length - (this.page * BOOKS_PER_PAGE), 0);
  document.querySelector('[data-list-button]').innerHTML = `
      <span>Show more</span>
      <span class="list__remaining"> (${remaining})</span>
  `;
}
```

---

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

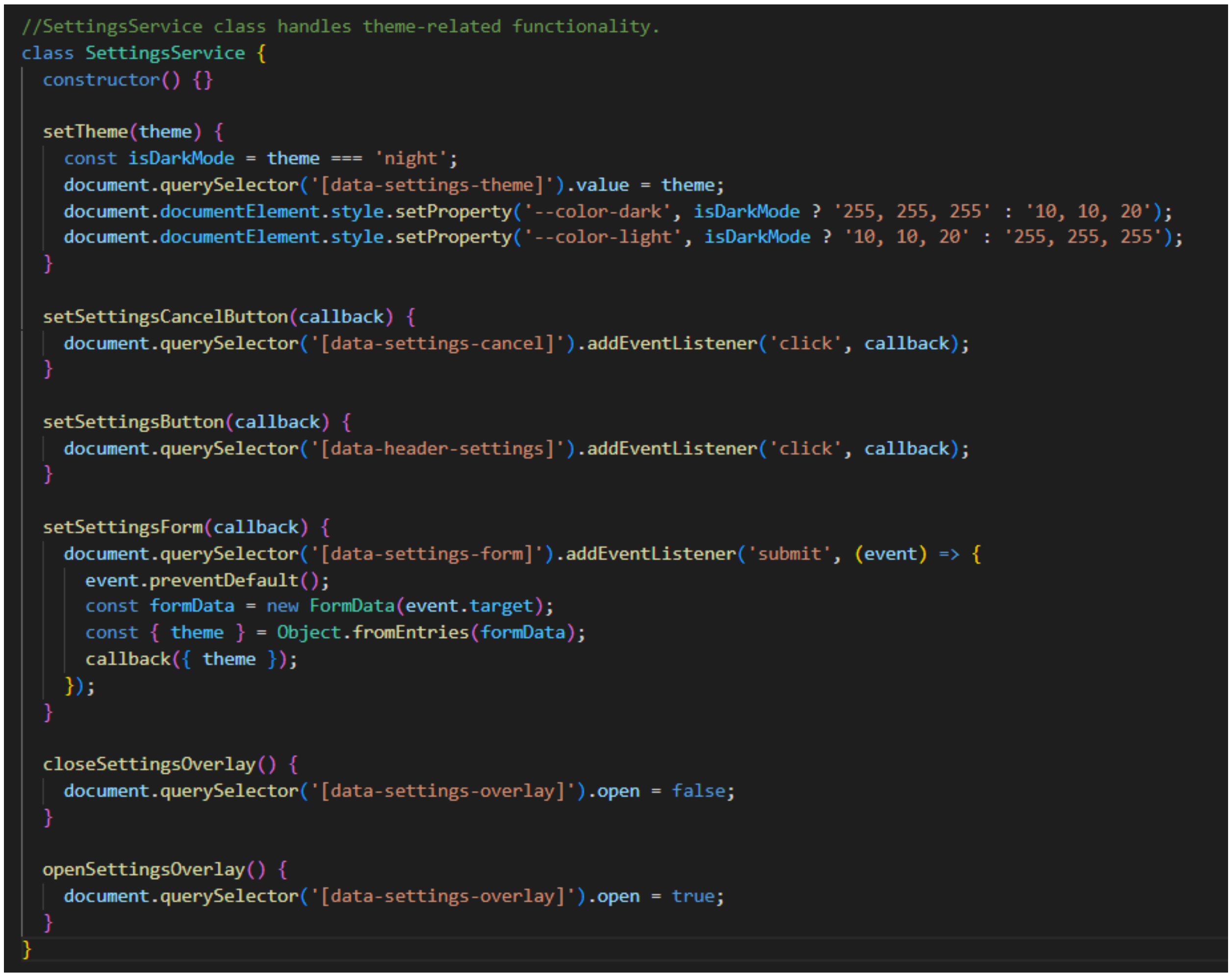
1.



```
//PreviewService class handles the creation and manipulation of preview elements.
class PreviewService {
  constructor(books, authors) {
    this.books = books;
    this.authors = authors;
  }

  createPreviewElement({ author, id, image, title }) {
    const element = document.createElement('button');
    element.classList = 'preview';
    element.setAttribute('data-preview', id);
    element.innerHTML = `
        <img
            class="preview__image"
            src="${image}"
        />
        <div class="preview__info">
            <h3 class="preview__title">${title}</h3>
            <div class="preview__author">${this.authors[author]}</div>
        </div>
    `;
    return element;
  }

  populatePreviewItems(startIndex, endIndex) {
    const fragment = document.createDocumentFragment();
    for (const book of this.books.slice(startIndex, endIndex)) {
      const element = this.createPreviewElement(book);
      fragment.appendChild(element);
    }
    return fragment;
  }

  appendPreviewItems(fragment) {
    document.querySelector('[data-list-items]').appendChild(fragment);
  }

  clearPreviewItems() {
    document.querySelector('[data-list-items]').innerHTML = '';
  }
```

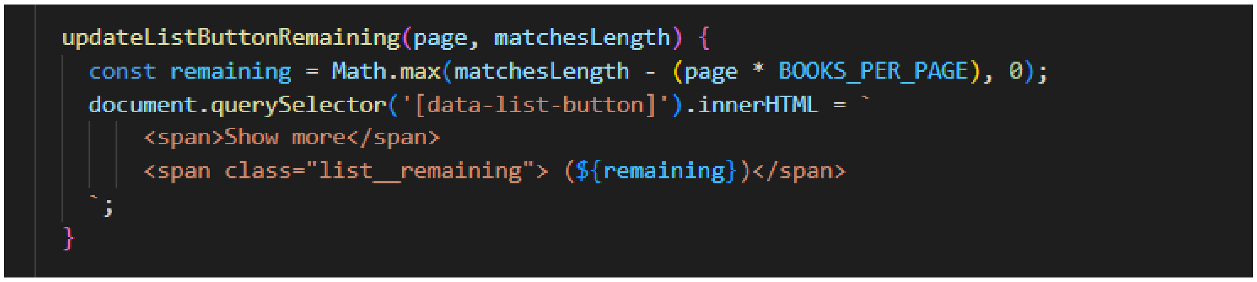
TB Mothibi_DWA7

2.



```
//SettingsService class handles theme-related functionality.
class SettingsService {
  constructor() {}

  setTheme(theme) {
    const isDarkMode = theme === 'night';
    document.querySelector('[data-settings-theme]').value = theme;
    document.documentElement.style.setProperty('--color-dark', isDarkMode ? '255, 255, 255' : '10, 10, 20');
    document.documentElement.style.setProperty('--color-light', isDarkMode ? '10, 10, 20' : '255, 255, 255');
  }

  setSettingsCancelButton(callback) {
    document.querySelector('[data-settings-cancel]').addEventListener('click', callback);
  }

  setSettingsButton(callback) {
    document.querySelector('[data-header-settings]').addEventListener('click', callback);
  }

  setSettingsForm(callback) {
    document.querySelector('[data-settings-form]').addEventListener('submit', (event) => {
      event.preventDefault();
      const formData = new FormData(event.target);
      const { theme } = Object.fromEntries(formData);
      callback({ theme });
    });
  }

  closeSettingsOverlay() {
    document.querySelector('[data-settings-overlay]').open = false;
  }

  openSettingsOverlay() {
    document.querySelector('[data-settings-overlay]').open = true;
  }
}
```

3.



```
  updateListButtonRemaining(page, matchesLength) {
    const remaining = Math.max(matchesLength - (page * BOOKS_PER_PAGE), 0);
    document.querySelector('[data-list-button]').innerHTML = `
        <span>Show more</span>
        <span class="list__remaining"> (${remaining})</span>
    `;
  }
```