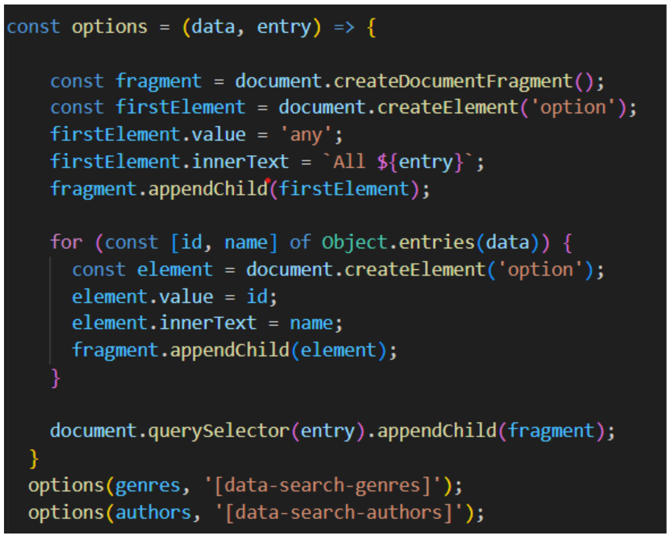
DWA_07.4 Knowledge Check_DWA7

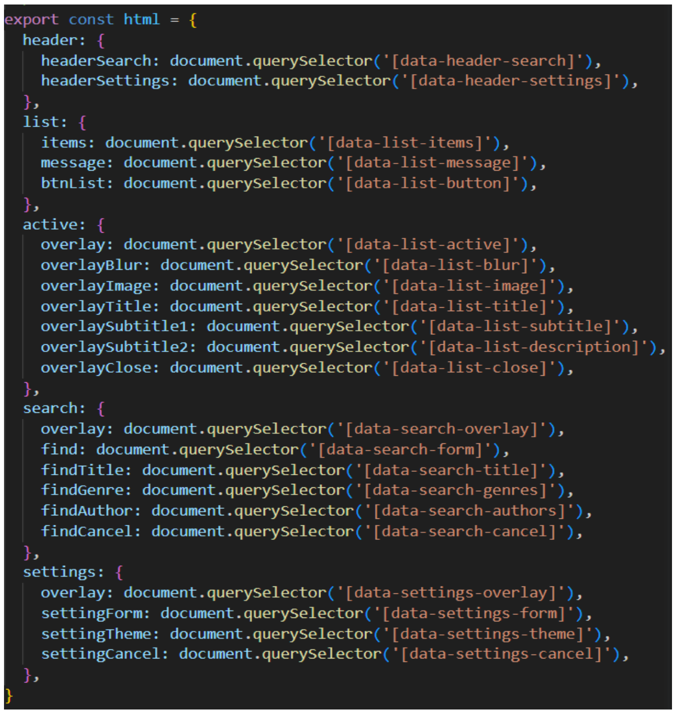
---

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



```
const options = (data, entry) => {

    const fragment = document.createDocumentFragment();
    const firstElement = document.createElement('option');
    firstElement.value = 'any';
    firstElement.innerText = `All ${entry}`;
    fragment.appendChild(firstElement);

    for (const [id, name] of Object.entries(data)) {
      const element = document.createElement('option');
      element.value = id;
      element.innerText = name;
      fragment.appendChild(element);
    }

    document.querySelector(entry).appendChild(fragment);
  }
  options(genres, '[data-search-genres]');
  options(authors, '[data-search-authors]');
```

Function where it creates a dropdown box for both the authors and the genre

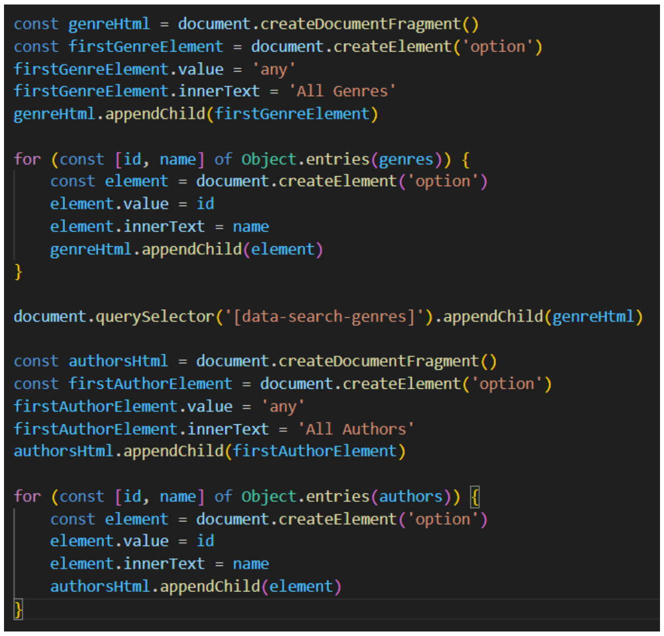


```
export const html = {
  header: {
    headerSearch: document.querySelector('[data-header-search]'),
    headerSettings: document.querySelector('[data-header-settings]'),
  },
  list: {
    items: document.querySelector('[data-list-items]'),
    message: document.querySelector('[data-list-message]'),
    btnList: document.querySelector('[data-list-button]'),
  },
  active: {
    overlay: document.querySelector('[data-list-active]'),
    overlayBlur: document.querySelector('[data-list-blur]'),
    overlayImage: document.querySelector('[data-list-image]'),
    overlayTitle: document.querySelector('[data-list-title]'),
    overlaySubtitle1: document.querySelector('[data-list-subtitle]'),
    overlaySubtitle2: document.querySelector('[data-list-description]'),
    overlayClose: document.querySelector('[data-list-close]'),
  },
  search: {
    overlay: document.querySelector('[data-search-overlay]'),
    find: document.querySelector('[data-search-form]'),
    findTitle: document.querySelector('[data-search-title]'),
    findGenre: document.querySelector('[data-search-genres]'),
    findAuthor: document.querySelector('[data-search-authors]'),
    findCancel: document.querySelector('[data-search-cancel]'),
  },
  settings: {
    overlay: document.querySelector('[data-settings-overlay]'),
    settingForm: document.querySelector('[data-settings-form]'),
    settingTheme: document.querySelector('[data-settings-theme]'),
    settingCancel: document.querySelector('[data-settings-cancel]'),
  },
}
```

All the data attributes are in one object where the can be called via EventListeners

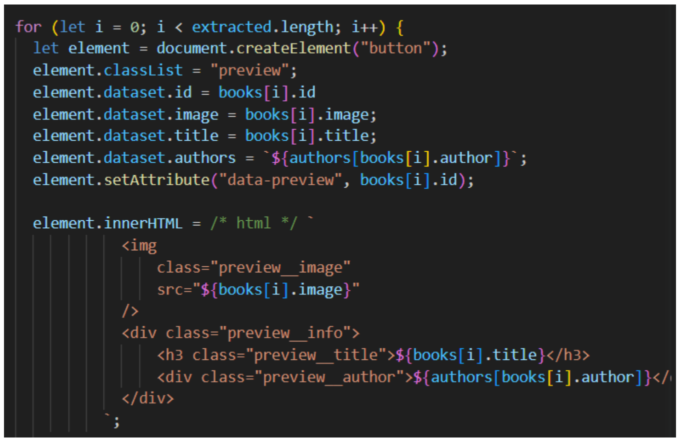
---

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



```
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)
}

document.querySelector('[data-search-genres]').appendChild(genreHtml)

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)
}
```

Code that creates a dropdown drop for both authors and genres which the makes the code to repeat the same thing but the output is different



```
for (let i = 0; i < extracted.length; i++) {
  let element = document.createElement("button");
  element.classList = "preview";
  element.dataset.id = books[i].id
  element.dataset.image = books[i].image;
  element.dataset.title = books[i].title;
  element.dataset.authors = `${authors[books[i].author]}`;
  element.setAttribute("data-preview", books[i].id);

  element.innerHTML = /* html */ `
          <img
              class="preview__image"
              src="${books[i].image}"
          />
          <div class="preview__info">
              <h3 class="preview__title">${books[i].title}</h3>
              <div class="preview__author">${authors[books[i].author]}</
          </div>
        `;
```

Code where it create an element where by the it uses dataset to get the required information but I have to code each dataset individually to get the desired information

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

- **Open-Closed Principle:** This will improve my code so if anything needs to be added, it can without modifying the code.
- **Liskov Substitution Principle:** If my object books/if any specific code is replaced, it should still execute the same way