CSC309

Home Syllabus Lectures Exercises & Project MarkUs
Piazza Anonymous Feedback

Programming on the Web

Exercise 2: Library System

Due Date: Tuesday October 23, before 11pm

Exercises are to be done **individually**, and are worth 5% of your course grade.

Updates

- **Sept 14:** In the event listeners around line 84, one of the lines should read `bookInfoForm.addEventListener('submit',` **`getBookInfo`**`)` , rather than **`displayBookInfo.`**
  The starter code has been updated.

- **Sept 14:** For `<table>` elements, the `<thead>` and `<tbody>` elements are created autmoatically in the DOM as direct children of `<table>`. You will have to work with them despite our starter HTML not having them there.
  The starter code has **not** been updated, but the handout section explaining it has been.

Reminder: GitHub IDs

Please add your GitHub ID to the spreadsheet in our latest announcements by **Wednesday, October 17**.

Overview:

In this exercise, you will using vanilla JavaScript to create a simple library book circulation system.

The goal is to build and edit DOM elements, while using built-in JS data structures (like objects and arrays) to store the data that these DOM elements display.

Library System

Our library system is simple: It has a collection of **books**, a collection of people who can loan books (known as **patrons**), and some interactions that allow books to be loaned and returned to the library.

Each book is uniqiuely identified by an ID number and each patron has a unique library card number.

Starter Files

Download the starter files **(updated Sep. 14)**.

The `e2_library.html` and the `e2_style.css` include the layout for this exercise. You **should not** change them. Your task is to add to the `e2_library.js` JavaScript file. The comments in the JS file will help guide you in completing the file correctly.

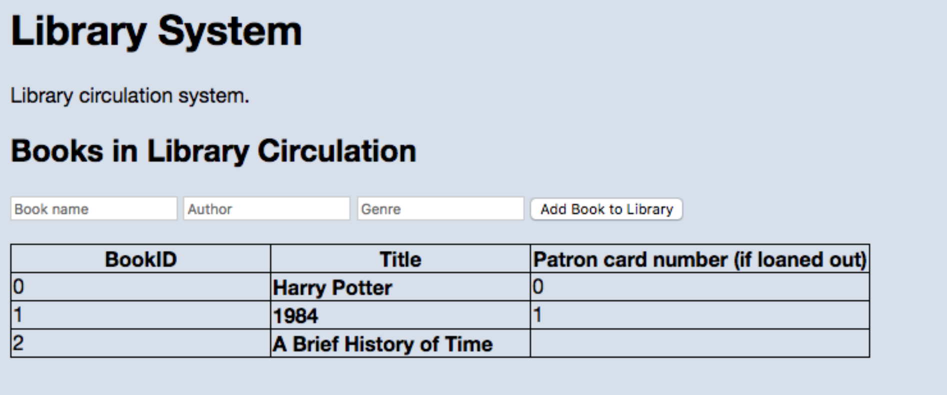
Do not change the names of the classes and id's in the starter files. They are used to select the proper elements in the JS code and to style elements properly in the CSS.

At the top of `e2_library.js`, you will see the data structures that will hold the data while the webpage is displayed in your browser (remember that refreshing will reset the data).
The `libraryBooks` and `patrons` global arrays will hold collections of `Book` and `Patron` objects that represent the books and patrons in the system.
Read all of the starter code and make sure you understand the intention behind each part.
Refer to our lecture notes if you're stuck.

Library System Specification

This section describes how the library system should work. The different sections of the page are explained.

1. Books in circulation: This table shows the books that the library has in circulation, including their book ID, title, and to whom they are loaned out (if applicable).



Library System

Library circulation system.

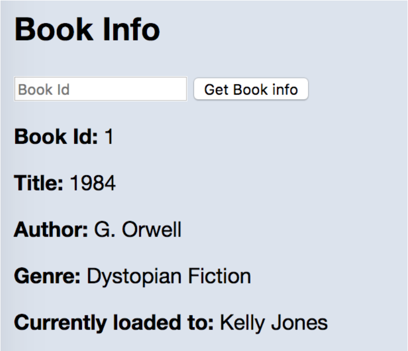
Books in Library Circulation

BookID	Title	Patron card number (if loaned out)
0	**Harry Potter**	0
1	**1984**	1
2	**A Brief History of Time**	

Note 1: BookID 1 is loaned out in the image above but not in starter HTML for demonstation purposes.
Note 2: Sometimes, the html file will show that a `<table>` element contains `<thead>` and `<tbody>` elements as direct children. ~~They are not included in our layout to make the DOM manipulation a little easier~~ **they are created automatically in the browser regardless, so you'll have to work with them accordingly.**

When the page first loads, some books and patrons will already be loaded into the JS global arrays, and will appear in the DOM (find where this happens in the JS file). This shows you how each entry should appear in the tables on the page, and lets you access the data for these entries in the JS global arrays. They are in the same order in the global arrays as the order of their IDs or library card numbers (i.e., the Book with ID 2 will be at index 2 in the `libraryBooks` array).

2. Book Info: This section displays all info about a book, including its book ID, title, author, genre, and the name of the patron it is loaned out to (if applicable).



Book Info

Book Id: 1

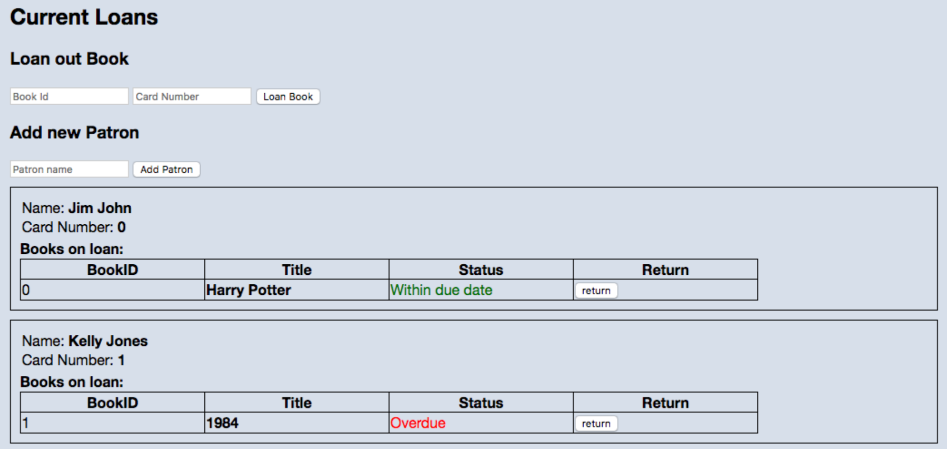
Title: 1984

Author: G. Orwell

Genre: Dystopian Fiction

Currently loaded to: Kelly Jones

3. Patrons and Loans: This section shows the patrons and the books they've loaned, and allows you set a book to be loaned by a patron.



Current Loans

Loan out Book

Add new Patron

Name: **Jim John**
Card Number: **0**
Books on loan:

BookID	Title	Status	Return
0	**Harry Potter**	Within due date	return

Name: **Kelly Jones**
Card Number: **1**
Books on loan:

BookID	Title	Status	Return
1	**1984**	Overdue	return

Patron Info: This section displays information about patrons and the books they have loaned out. Each patron has a library Card Number.

The 'Books on loan' table for each patron shows the bookID, title, author, and the *status* of the book loan.
The status can be one of two values: Within due date, or Overdue.
How do we tell if a book is overdue? When a book is loaned out, the status is set as a green Within due date. We then immediately start a timer for 3 seconds (representing 3 days, or the loan period for a book) from the time of loan. Once these 3 seconds is up, the value should change to a red Overdue.

'return' button: Clicking the return button beside a book returns the book to the library, removing the patron's card number from the book's entry in the top circulation table.

Your Tasks

In order to implement the library system, you must fill in the functions in the `e2_library.js` file.

The comments in this file give tips on what each function should do, and in what order it should be done. It is up to you to figure out how make them work properly.

The first several functions are callbacks to the event listeners (functions that are run when 'submit' or 'click' events are run). They are meant to change the JS data structures, and then call on the function that manipulates the DOM (under the comment `/* DOM functions below */`).
Only manipulate the DOM (create/edit/remove HTML elements) in these last six functions.

They should all work together to make the library system work.

Use the html elements given in the starter code as templates for the elements you have to make in your JS functions. They must look exactly as given, so build up the DOM elements accordingly.

Console Log

Remember that you can debug your JavaScript in the Developer Tools console. Log variables to the console to make sure your program is calling the correct functions.

Important Information: Please read!

- You can assume that input will be valid - you do not need to check for errors or non-existant books/patrons, or if books were already loaned out.
- Do **not** change the beginning of the e2_library.js starter code. You may only edit the parts of the JavaScript where indicated. Do not change the definitions of the object/constructor prototypes.
- Do **not** add or remove properties of the Book or Patron objects/prototypes. You can complete this exercise with the ones given.
- Do **not** change the HTML or CSS files. You may only edit the parts of the JavaScript where indicated. Do not change the definitions of the object/constructor prototypes.
- You may **not** import or use any external frameworks or libraries. You can only (and only need to) use vanilla JavaScript, including techniques we talked about in class and the lab. Using external tools (or copying content from those tools into your code) will result in a grade of **0**.
- We will test on **Google Chrome**, so make sure it works in that browser.

Grading

We will grade your submission on the following criteria:

- Functions correctly create and edit JS data structures (objects and arrays).
- DOM manipulation is correct - elements are correctly added/edited/removed when an action is performed.

What to submit

Edit the given `e2_library.js` file to make the Library system work.
Make sure there are **no errors** in the console when loading the page or when running your functions. Submit this file on MarkUs (you do not need to submit the HTML and CSS files, since you will not be changing them).