

Lab 4: JavaScript in the Browser

In this lab, you will add dynamic behaviors to the FilmLibrary interface that you have implemented in the last lab.

1. Populate the Film Library

Starting from the outcome of the previous lab, replace the hard-coded films displayed in the main content area with a dynamically populated list of films. For this purpose, use the same **FilmLibrary** object you created in Lab 1, containing the films with their corresponding data. Remember that this **FilmLibrary** object must be declared in a JavaScript file that you should import into the HTML file.

Then, modify the GUI to display the films in the list dynamically. To ensure the correct integration of the **FilmLibrary** object, if you change the content in its JS Films array and reload the page, the list in the main content area must update accordingly.

2. Let's make the filters work!

Implement the filtering functionality by associating the items in the left sidebar to the following filters:

- **All:** retrieves and display *all* the films in the Film Library.
- **Favorite:** retrieves and display only films marked as *favorite*.
- **Best Rated:** retrieves and display only films whose score is **five** out of five.
- **Seen Last Month:** retrieves and display the films watched between today and the last 30 days.

Use the DOM manipulation methods to **update** the list of films displayed in the webpage based on the applied filter. When a filter is applied, you must update the specific component where the films are being displayed.

Take also into account that the filtering function should not modify the original set of Films in the JavaScript data structure, it should just return a subset of them. Finally, add some suitable entries to the **FilmLibrary** to test all the filters.

Note: One and only one filter may be active at a time. As specified in the previous laboratory, "**All**" is the active filter by default. The interface should always highlight the currently applied filter.

Additionally, the interface must display, at the top of the main content area, the name of the current applied filter.

3. Optional: Add the functionality to delete the Films

Enable users to delete specific films. Add a trash icon (🗑️) next to the Films' title and link it to a JS function that removes the corresponding Film from the **FilmLibrary** array. When a Film is deleted, the list of films should update automatically, maintaining the current filter activated.

Hints:

1. Use the Bootstrap 5 framework and its components:
<https://getbootstrap.com>
2. Use the methods for DOM manipulation to dynamically populate the list of films defined in the JavaScript data structure (e.g., `getElementById`, `createElement`)
3. You can use the following script tag to import and use the day.js library:
`<script defer src="https://unpkg.com/dayjs@1.8.21/dayjs.min.js" />`
(see the details on <https://day.js.org/docs/en/installation/browser>)
4. You can use the available Lab 3 solution as starting point:
<https://github.com/polito-WA1-AW1-2022/lab3-html-css>