Exercise04: Javascript

Objectives:

To learn to use **JQuery**, **JSON**, and **Object-oriented Javascript**.

Work with your group (or by yourself). Each group is to upload only one submission.

Credits (for edits): Zahra Hosseini ,Naresh Somisetty, Isaac Martin.

EXAMPLES

00: show the order in which js is loaded and executed and ready and onload events are fired.

01: shows self-invocation of functions. Also, shows use of call, apply, and bind.

02: shows JSON stringify and parse methods.

03: Shows adding action with Jquery to DOM

03-2: Shows adding action with Jquery to DOM via id attribute, tag and class

03-3: create table with Jquery (i.e. use jquery to create DOM elements)

04: two ways to create objects

05: factory pattern of creating objects and why it does not work.

06: shows constructor pattern for creating js objects and the problem with that.

07: shows the prototype pattern for creating js objects and the problem with that.

08: shows the constructor+prototype approach of creating js objects.

1. Warm Up: Try Some Examples

1. First, open blackboard, go to Course Contents, and then download exercise06.zip file into your workspace (U:\workspace or something like that!). Then, unzip.

2. Play with each of the given examples (in examples directory). Open them using a text editor of your choice and modify parts of the html or js files.

Please do the TODO segments for each example.

You will need to also learn how to use the available tools for JS debugging.
Safari has Develop menu with "show error console" etc, Firefox has tools->WebDeveloper->Debugger, Chrome has Tools->Developer Tools.

IT IS REQUIRED THAT YOU TRY EACH EXAMPLE AND DO THE "TODO" SECTIONS.

3. ADDITIONAL RESOURCES

Please read the concepts in below link. And also try examples provided for best practices.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript#Object-oriented_programming

JQuery Documentation and Examples. https://learn.jquery.com/using-jquery-core/ https://learn.jquery.com/about-jquery/, also w3schools.com

2. JQuery

This is a really simple exercise to have you practice jquery. Create an HTML page (that's empty for now) that you'll create/manipulate as you explore. Add code to load the jQuery library and a code.js JavaScript file that you'll also create.

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script src="code.js"></script>
<script>
  // page-specific code goes here
</script>
```

Add code to the code.js file that demonstrates jQuery features. Create a small demo of each of following jQuery features. Obviously, you will have to create and add HTML content so you can see your demos working.

- Write simple comment for each effect, like this
  <!-- Html Element: button id:..effect:…-->
- Manipulate the CSS (i.e. style) of an element with jQuery. Show FIVE different style elements being changed.
- Use an effect on an element. Show FIVE different effects. http://learn.jquery.com/effects/.
- Use jQuery events. Show FIVE different events. http://learn.jquery.com/events/

You will need to submit your code.html and code.js files. Attach every required file and remove unrelated files from your submission.

GO TO NEXT PAGE

3. Library

The goal of this exercise is to design and implement a books library using object-oriented Javascript. Use a constructor+prototype pattern to develop your objects (look at examples to see what we mean). At a minimum, your design should have the three classes Library, Shelf, and Book. Write all the code for these objects in booksLibrary.js file.

Assume we have the following types of book in the Library.

- 5 Reference books
- 20 ordinary books.

Assume each book has an ID.

There are four categories of books : Art, Science, Sport, literature. you should categorize book based on this categories. The algorithm for categorizing should be like following

If (BookID%4==0) categorize book as Art

if (BookID%4==1) categorize book as science

If (BookID%4==2) categorize book as Sport

if (BookID%4==3) categorize book as Literature

Note: Different category should be in different shelves.

Each book should have an attribute of borrowedBy which showed the user name of student who borrowed that book. At the first none of books are borrowed by students.

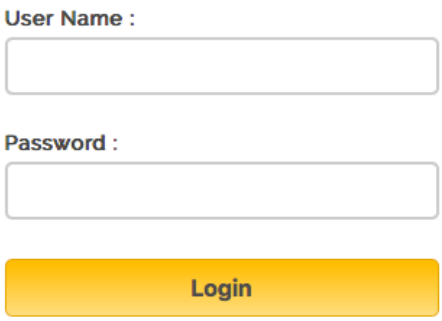
Each book should have an attribute of presence which shows the presence (1) or borrowed(0) situation of the book. At the first all books are presence.

Use local storage in order to save presence attribute and borrowedBy attribute for each book.

Note that Reference books cannot be checked out.

The library System Operations:

1. Login page:



1. User Interface: Should consist of Username and password fields and the login button.

2. Functionality:

    1. If the Username and password are equal to admin, login as a librarian.

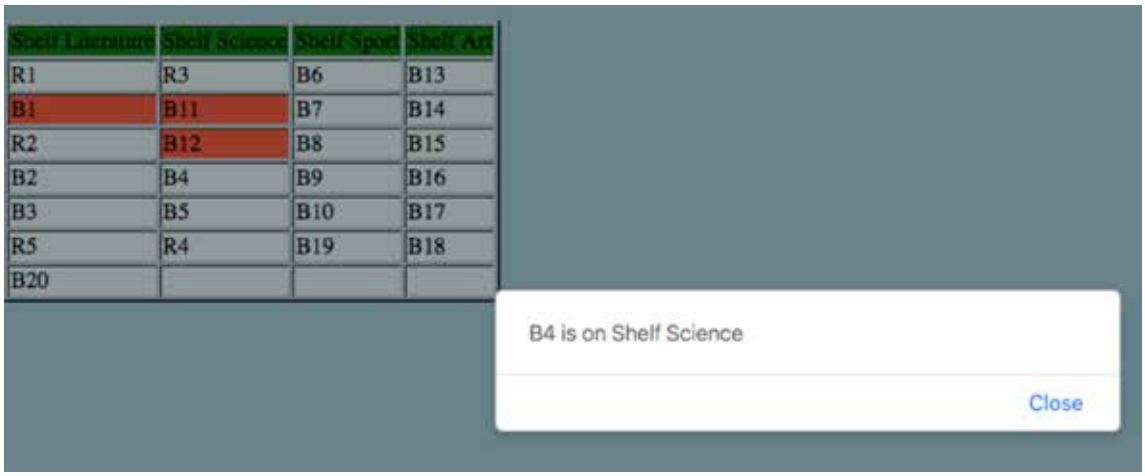
    2. If the username start with U, login as an Undergraduate student.

    3. Otherwise, Show an alert that this is not a correct username and password.

2. Librarian view:

Shelf Literature	Shelf Science	Shelf Sport	Shelf Art
R1	R3	B6	B13
B1	B11	B7	B14
R2	B12	B8	B15
B2	B4	B9	B16
B3	B5	B10	B17
R5	R4	B19	B18
B20			

1. User interface: Show a visual representation of the library with current shelves and books. HTML table is to be generated dynamically where each column is a shelf. Each cell is a book.



2. Functionality:

1. Ability to add specific book to specific shelf.

2. clicking on the cell gives book's detail.

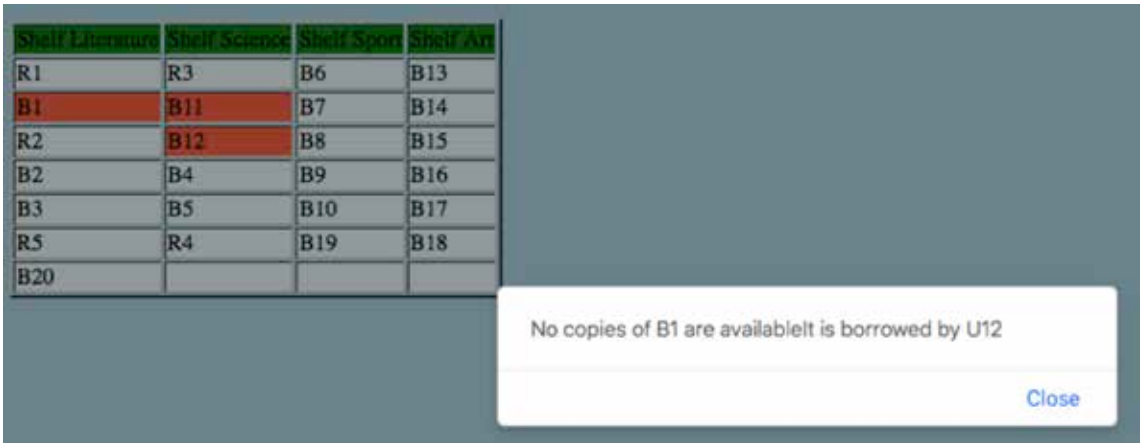
3. Undergraduate View:

1. UserInterface: Undergraduate student should have the similar view as a librarian.

2. Functionality:

1. Undergraduate student can borrow at most two books each time. So, a student can select on at most two books and change the color of that cell to red (i.e. the book is borrowed, so change the borrowedBy attribute to its username).

2. Student should be able to return the book which he/she borrowed.



MAKE SURE TO:

- Write a README.txt file – to describe what you did. Will help in grading.
- Use Constructor+Prototype pattern to create objects.

- It should contain minimum three classes Library, Shelf, Book.
- Don’t use any global variables or global functions.
- You should decide the members and operations of each class.
- Make sure to have the Library object perform the required operations.
- Write a Client HTML code that includes the booksLibrary.js file and uses it (i.e. create library objects and populate them and display them etc).
- **You will need to submit your booksLibrary.js and booksLibrary.html files.**

Submission:

Make sure your solutions work on Chrome (which is what TAs will use to grade the assignment). Zip your html, js files, and participation file (i.e. who worked on which part or if you worked together). Then, submit this zip file on black board. Remember there is only one submission per group. Make sure to include all the files that are needed in order to run your program.

Participation file is a simple txt file, which clarifies the specific participation of two members.