JavaScript Web Storage Exercise

In this exercise you will be required to create an object many times. Each instance of the object will be stored locally initially and then stored by the web session. The object will be converted to JSON format, before adding it to storage. Once the objects are stored, then they will need to be accessed. This will involve using the JSON object to convert the JSON formatted stored value to a JavaScript object. At this point, the object property for each instance can be checked to verify the expected objects were held in storage.

Exercise 1 – Create an Object, and store instances of that object locally

The steps outlined below will help you complete the exercise:

1. On moodle download the zip file included with this exercise and extract the details to a location of your choice on your device.
2. In the js folder, update local-storage.js so that it includes a function named book. This function should include a parameter for name, which is used to set the name property of the book object. Refer to JavaScript Objects Exercise 2 for as a reference for creating such a function.
3. Update local-storage.js to include an add function. This function is invoked by clicking the Add button in local_storage.html.
4. The add function must do the following:
    a. Create a unique key value, which will be used for storing the object instance. You could use a static string appended to a unique value such as the storage length property for this purpose.
    b. Invoke the book function to read the value entered in the text field in local_storage.html to create an instance of the book object.
    c. Convert this instance of the book object to JSON using the JSON objects stringify method (JSON.stringify()).
    d. Use the localStorage objects setItem method to add the key, as defined in step a, and the value, as defined in step c, to local storage.
    e. Add the following line of code as the last line in the function to reload the page so that the web page is refreshed:

       ```
       window.location.reload();
       ```
5. Save all changes and open the local_storage.html in a web browser – preferably chrome.
6. Type some text into the text field and click the Add button. Then type different text into the text field and click the Add button. This should ensure you have two keys, with JSON formatted values saved to the local web storage. Exercise 3, will test for this, as additional javascript is required to retrieve these values.

Exercise 2 – Clear Web Storage

The steps outlined below will help you complete the exercise:

1. Update local-storage.js to include a clear_storage function. This function is invoked by clicking the Clear Storage button in local_storage.html.
2. Add a line of code to this function to invoke the localStorage Objects clear method.
3. Save changes, and DO NOT click on the Clear Storage button, till exercise 3.

Exercise 3 – Access Local Storage

The steps outlined below will help you complete the exercise:

1. Update local-storage.js to include a fromStorage function. This function is invoked by clicking the Cart button in submit.html.
2. The fromStorage function must do the following:
    a. Include iteration, e.g. for loop, to access each key/value from the local storage. The length property can be used as a value in the iteration condition.
    b. Within the iteration, access relevant key applicable to the current iteration, using localStorage objects getItem method.
    c. The value returned will be in JSON format, so convert it to a javascript object using the JSON.parse() method. Pass the JSON formatted value to this method.
    d. For the last line within the iteration, use an alert, to display the name property of the object.
3. Save changes to local-storage.js.
4. Reload local-storage.html.
5. Click on the Cart button.
6. The submit.html form should appear.
7. Then click on the Cart button.
8. Confirm that two alerts appear showing the name values entered in the text field in local_storage.html page in step 6 of exercise 1.
9. Reload local_storage.html and click the Clear Storage button.
10. Then click the Cart button.
11. The submit.html form should appear.
12. Then click on the Cart button.
13. Confirm that no alerts appear.
14. Reload local_storage.html and add some more objects to local storage.
15. Exit the browser session, reopen the browser, load local-storage.html, click the cart button, and then click the cart button on the page which appears. Confirm alerts display showing details of the objects you entered in step 14. As local storage is used, this data will remain available unless the clear method on the localStorage object is invoked.

Exercise 4 – Session Storage

The steps outlined below will help you complete the exercise:

1. Copy the html files and the js file used in exercises 1 to 4 to a new location.
2. Open local_storage.js, and wherever localStorage object is found, replace it with sessionStorage object.
3. Save the changes to local_storage.js.
4. Open local_storage.html and add some objects to session storage, by entering values in the text field and clicking the add button.
5. Then click on the cart button, and again on the cart button when the submit.html web page appears.
6. Confirm that an alert appears showing the names of the objects you added in step 4.
7. Exit the browser session, reopen the browser, load local-storage.html, click the cart button, and then click the cart button on the page which appears.
8. Nothing should happen, as in no alerts should appear. By exiting the browser in step 7, this should have clear the session storage.