Intro to Client-Side Storage

The Coding Bootcamp

Class Objectives

- Introduce the concept and utility of "data persistence" in web applications

- Demonstrate different methods for client-side data storage (i.e. localStorage, sessionStorage, and cookies)

- Build a semi-complex application taking advantage of localStorage to maintain content between browser sessions.

Get into groups!

- Assign one student of the three of you to be the group's typist. All others in the group will help supervise, offer guidance, and help catch bugs that the group's typist might not be able to catch. Throughout this activity (and many of today's activities) you will be switching roles with those in your group.

- Then create a Basic To-Do application using the `1-student-do-todolist.html` as a starting point. The HTML is already created for you so you will only have to fill in the javascript sections where there are large blanks. Alternatively, if you are feeling bold and capable, your group may code it from scratch without the use of the base code provided.

- Helpful Hint: Each of the buttons and text elements should have a dynamically created identifier or data attribute that differentiates one item on the list from another. You should use these attributes to close out the appropriate items when you click on the "✓" beside them.

Client vs. Server Storage

We can temporarily store data on users' computers + browsers (client-side) and we can also store that information on an external server/database (server-side).

Today we cover client-side storage

Sign-in Page Without Persistence

- Using the `2-student-do-signin-no-persistence` as a starting point, fill in the JavaScript code necessary to make the page "save user inputs" and then re-display them on the second panel (most recent member).

- Note: Don't worry about using client-side saving just yet. Just focus on getting the text inside the inputs and then displaying them via html in the second panel.

Sign-in Page with localStorage

- Using the solution provided to you in `2-student-do-signin-no-persistence-solution.html`, re-configure the application so that it utilizes localStorage.

- If your code worked it should save/display the last inputted user even if the tab is closed or if the page is closed and reopened.

sessionStorage

Client-side Storage

LocalStorage	5MB/10MB storage It's not session based, need to be deleted via JS or manually Client side reading only Less older browsers support
SessionStorage	5MB storage It's session based and working per window or tab Client side reading only Less older browsers support
Cookie	4KB storage Expiry depends on the setting and working per window or tab Server and client side reading More older browsers support

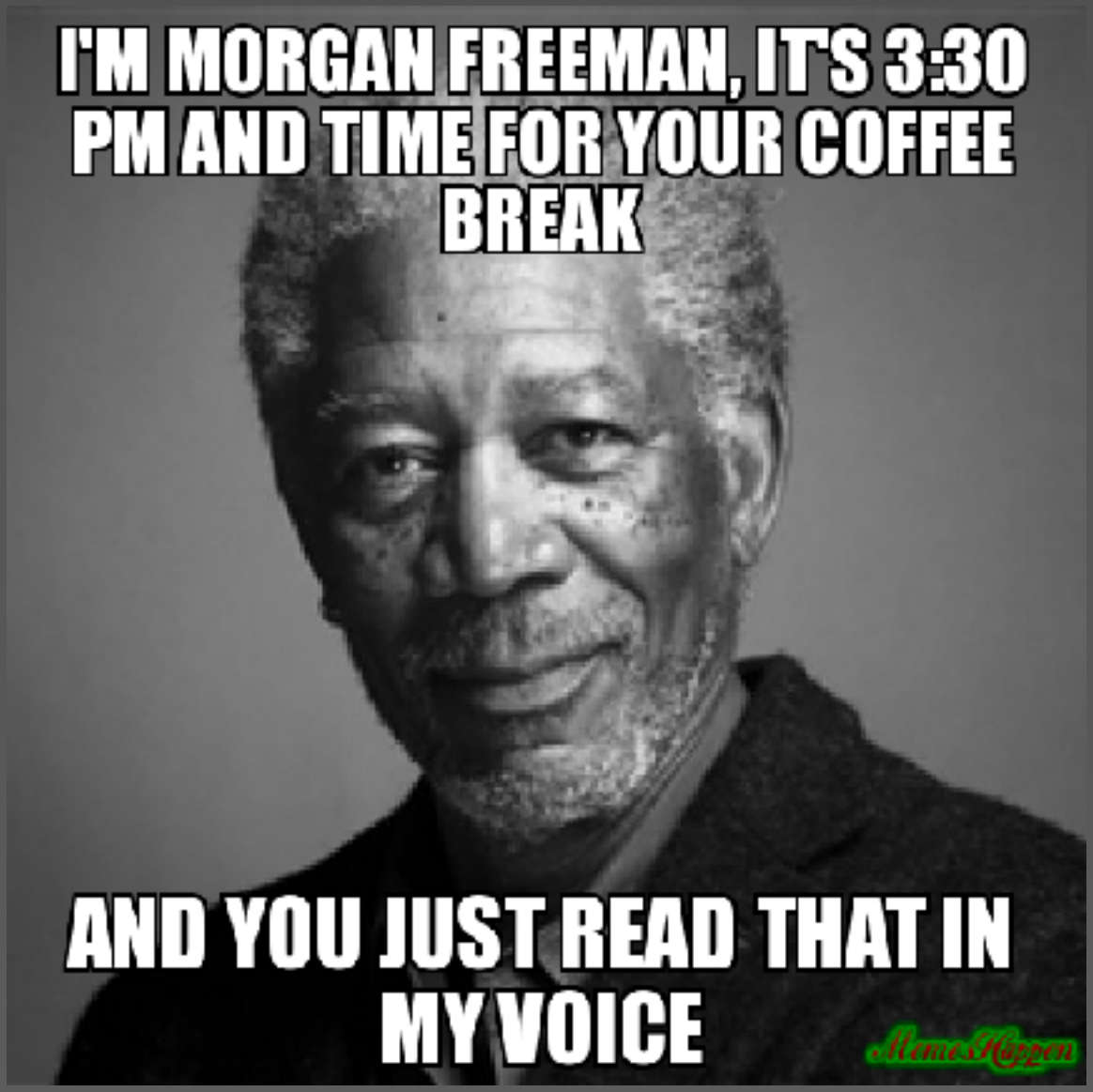
JSON in local/session storage

- **localStorage**.setItem('user', **JSON**.stringify(user));
  Then to retrieve it from the **store** and convert to an object again:

- var user = **JSON**.parse(**localStorage**.getItem('user'));

- If we need to delete all entries of the **store** we can simply do: localStorage.clear();

Break


I'M MORGAN FREEMAN, IT'S 3:30 PM AND TIME FOR YOUR COFFEE BREAK
AND YOU JUST READ THAT IN MY VOICE
MemeHappen

C is for Cookie, but not for Chrome

Chrome doesn't support cookies on static local non-deployed HTML pages. There are workarounds for this, but in order to demonstrate the activity you can use the firefox or safari browser to demonstrate the desired functionality if desired and convenient.

Cookies are not necessary for this weeks homework.

Partners Do: To-Do with localStorage

- Using a working example of the to-do application `1-student-do-todolist-solution.html` incorporate the use of `localStorage` to create data persistence.
- Have the group member who hasn't typed yet become the typist in the group while the other two take on the role of supervisor.
- HINT: You will need to create an additional array of todos that you can keep adding todo items to.
- HINT: You will need to selectively delete array elements to get this working properly. (Suggestion: Look into `.splice`)
- HINT: You will need to take an array and dump the contents into localStorage (Suggestion: Use `JSON.stringify(todoArray)`).
- HINT: Don't freak out. This is hard, but push yourselves as best you can!

Homework

Train Time

This app lets you see train schedules in the future based on the current time.

When you create a new train, you specify the start time and frequency and it will then calculate the next train on the schedule.

Intro to Firebase

https://www.firebase.com/docs/web/quickstart.html

Questions