Samuel Herscher

Dynamic Web 1

4/29/20

Questions week #9

1. The Web Storage API allowed applications to store data within an individual user's browser.
2. Before HTML 5 data had to be stored in cookies. The date storage was called Cookies and it was included in each server request.
3. 3 advantages of using Web storage over cookies is web storage is more secure as well as large amounts of data can be stored locally, the information is never transferred to the server.
4. The 2 objects are called localStorage object and sessionStorage objects.
5. No, localStorage object and sessionStorage can only be read on client-side.
6. The httpOnly flag blocks access can be read server-side. Cookies are popular because they allow sites to store user data directly into the user’s web browser. It’s mostly used to identify a user’s session because It might contain sensitive data.
7. Because Cookies could contain sensitive information like passwords or credit card information. Which is why they need to be secure and protected.
8. HttpOnly flag blocks access of cookie on the client side. So, if some sensitive info does leak into your JavaScript even though you have taking security measures it will not allow you to access those Cookies info on the client side.
9. The localStorage and sessionStorage of the web storage API are linked to a specific origin and domain and protocol. The Connection makes it possible for all pages that’s linked to one origin to store and access the same data.
10. The property is called the (Key) length property. More info
11. The Storage Object, represented by the localStorage and sessionStorage objects contains storage.length, the length property which returns an integer representing the number of data items stored in the Storage object, a Project application example could be: let aLength = storage.length;. and the Return value is : an integer. And lastly the Code example: function populateStorage() {localStorage.setItem('color', 'blue'); localStorage.setItem('font', 'Arial');
12. Method 1: Storage.key() - when it is passed a certain number represented by n, this method returns the name of the nth key in the storage. The order of keys is user-agent defined, so it's not reliable an Example is: console.log(localStorage.getItem(localStorage.key(i)));. Method 2: Storage.getItem() - when this method is passed a key name, it will return that key's value, or null if the key does not exist an example is: (sessionStorage.getItem('autosave', myStory)) Method 3: Storage.setItem() - when passed a key name and value, it will add that key to the storage, or update that key's value if it already exist an example is: (sessionStorage.setItem('autosave', myStory));. Method 4: Storage.removeItem() - when passed a key name, it will remove that key from the storage. If there is no item associated with the given key, this method will do nothing an example is: sessionStorage.removeItem('autosave', myStory.value);. And lastly method 5: Storage.clear() - when invoked, will empty all keys out of the storage an example is: function clearStorage() {const myStory = document.getElementById('textArea'); myStory.value = '';
13. You want to use the secure flag because you want to use the http0nly secure flag helps provent client-side access to the cookie.
14. The difference between localStorage and sessionStorage is that while data in localStorage doesn't expire and the data in sessionStorage is cleared when the page session ends.
15. The similarities between localStorage and sessionStorage are they both are used to store data on the client side and each one has its own storage and expiration limit.
16. “Try” allows you to define a block of code to be tested for any errors while it is being used. The “Catch” allows you to define a block of code to be executed if there is an error that occurs in the Try block.
17. You have to set a try...catch statement in the application in order to catch our exception. You set and get our sessionStorage and also check if we have exceeded our sessionStorage quota.
18. JSON.stringify() converts a JavaScript object or value to a JSON string, replacing values if a replacer function is selected. const csv = JSON.stringify(sessionStorage['autosave']);
19. The csv value is the result of the transformation of the javaScript object into a string and it’s important for your application so that we may insert the value, which is transferred into a string, into a .txt file for the user to download. It is also important because it lets you know whether the text has been saved or not.
20. I had a hard time trying to check what message I got :/