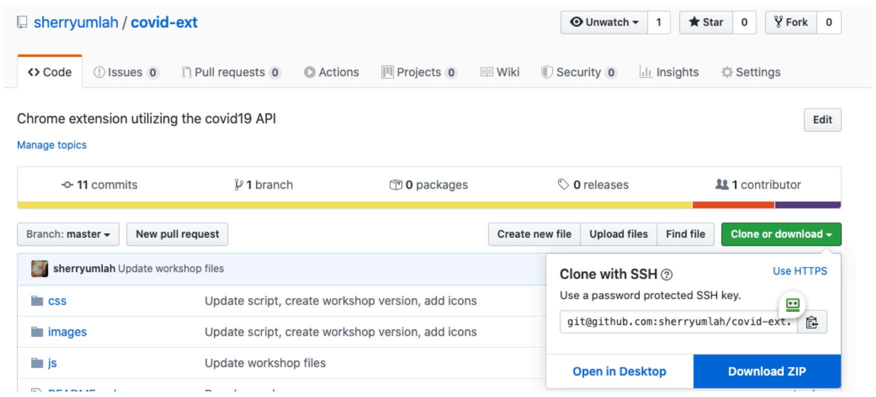
Covid-19 Chrome Extension Workshop Steps

Prerequisites:

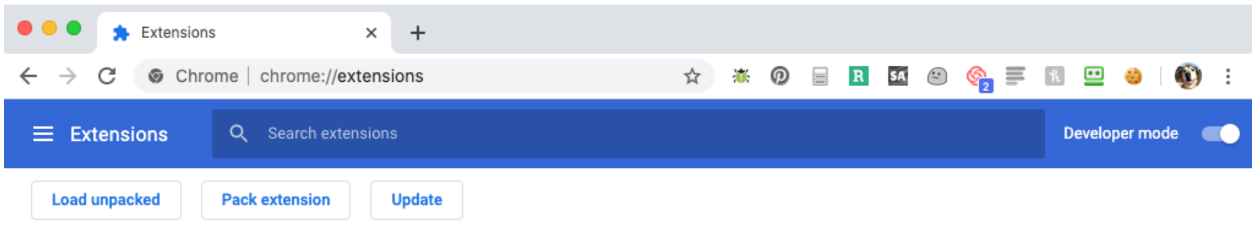
- Chrome Browser installed
- IDE/Code editor of your choice installed
- Basic understanding of HTML, CSS, and JavaScript
- Download or clone the repo at: https://github.com/sherryumlah.coom/covid-ext



Pre-Workshop Set-up Instructions:

Install the Extension:

1. After saving the github repository files to your local drive, type **chrome://extensions** in your browser's address bar and press the return key to navigate to Chrome's extensions page.



2. Ensure that the **Developer mode** toggle in the upper-right corner is toggled ON.

3. Click the **Load unpacked** button and select the extension folder that you saved to your drive in step 1.

4. Click **Select**.

The extension has now been installed. Each time you open a new tab in Chrome, this extension will display a photo of a random state with accompanying Covid-19 statistics for the state.

Workshop Instructions:

1. Rename script.js to script-final.js

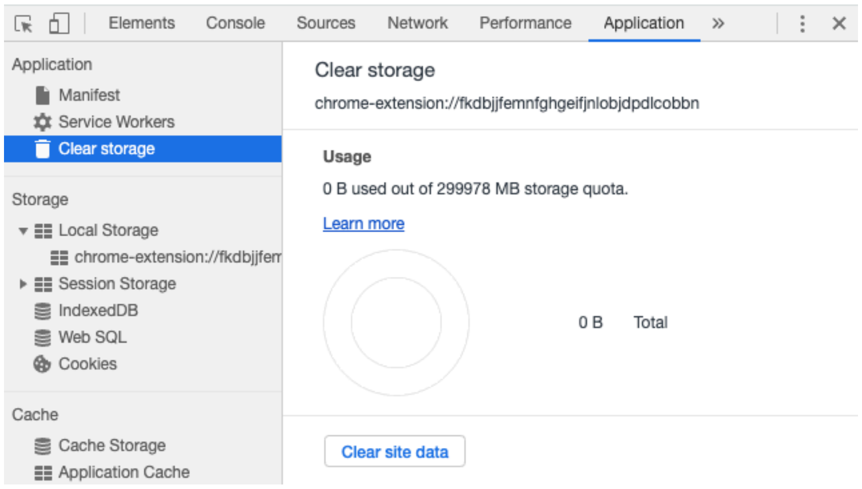
2. Rename script-wip.js to script.js

3. Open a new tab.

4. Open the **JavaScript Console.**
   This is accessible by right-clicking on the page and choosing **Inspect** and clicking on the **Console** tab or via the **View** menu:
   View > Developer > JavaScript Console.

5. Examine the console. Note that it only contains a log of today's date. Navigating to the **Application** tab, no key/value pairs should be visible in Storage > Local Storage.

   Note: If you have any values stored here from previously running the extension, clear them now by clicking on **Clear storage** under the **Application** section and then clicking the **Clear site data** button.



6. Uncomment line **124** and lines **39-42**, then save the file.
   This code will enable us to look for any previously stored values in local storage. Specifically, it looks to see if we have a value for the **dateLastRetrieved** key in local storage.

7. Refresh the page in your browser. The console log reads:
   Line 40: Covid API data last retrieved on: null.
   This value is null because we haven't fetched and stored data from the API yet.

8. Uncomment lines **125-132**. This block of code compare's today's date to any date value that might be stored in the **dateLastRetrieved** key in local storage.

   If the dates don't match, or the **localCovidData** key value is null or empty, we call the **getCovidData** function to retrieve data from the API and pass in today's date.

   If the dates DO match and the **localCovidData** key in local storage has data for its value, then we retrieve data from our local storage and display it.

   This block of code requires 3 functions: **getCovidData, displayData**, and **getLocalCovidData**. We'll need to uncomment those in the next few steps.

9. Uncomment lines **60-89** to make the **getCovidData** function available to us. Save the file and refresh the page in the browser.

   Note that the console suggests there's an error and that local storage has been cleared. This is because the **getCovidData** function relies on three other functions: **setLastRetrievedDate**, **getLocalCovidData**, and **displayData**. It's not finding those functions and erroring and we're catching the error and clearing the local storage to keep from saving bad or partial data.

10. Uncomment lines **44-47** to make the **setLastRetrievedDate** function available to us. This function takes a parameter for today's date and stores that value for the **dateLastRetrieved** key in local storage.

11. Uncomment lines **52-55** to make the **getLocalCovidData** function available to us. This function pulls the value for the **covidData** key from local storage and returns it.

12. Uncomment lines 103-120 to make the **displayData** function available to us. This function takes a parameter with the **covidData** key value from local storage and lets us parse it.

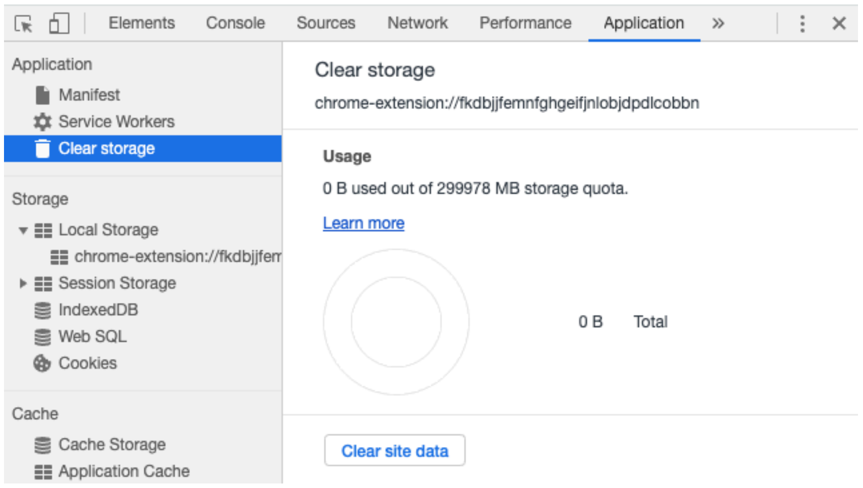
    We generate a random number and then use it to select an object from our array of stored API results. Then we write the object's properties inside elements on our newtab.html page, including the **Province** (state name), **Active** (active cases), **Deaths**, and **Date** properties.

    We also look for a file in the local images folder of the extension package that matches the name of our object's **Province** property.

13. The **displayData** function requires the **addCommas** function. Uncomment line **92**. This allows us to use the **addCommas** function, which takes a number as a parameter and returns the number properly formatted with commas. For example: 35070 is returned as 35,070.

14. Now that all of the code is uncommented, we can see the extension in action!

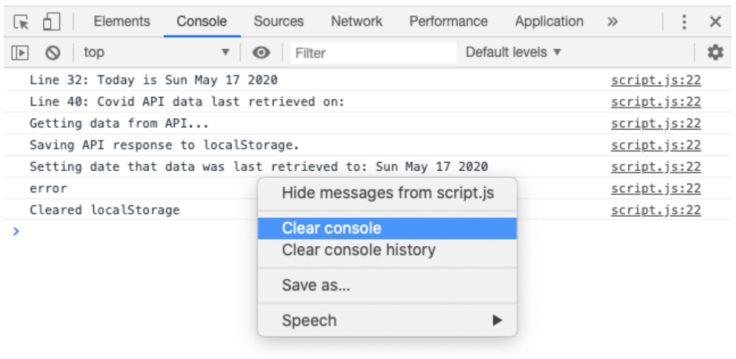
    First, be sure any previously stored values are cleared by clicking on **Clear storage** under the **Application** section and then clicking the **Clear site data** button.



chrome-extension://fkdbjjfemnfghgeifjnlobjdpdlcobbn

0 B used out of 299978 MB storage quota.

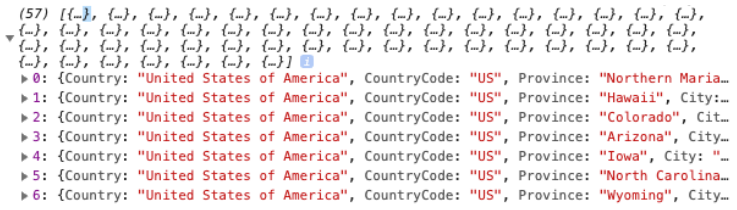
15. Clear the console by right-clicking in the console pane and choosing **Clear console** from the menu.



```
Line 32: Today is Sun May 17 2020                                  script.js:22
Line 40: Covid API data last retrieved on:                         script.js:22
Getting data from API...                                           script.js:22
Saving API response to localStorage.                               script.js:22
Setting date that data was last retrieved to: Sun May 17 2020      script.js:22
error                                                              script.js:22
Cleared localStorage                                               script.js:22
```

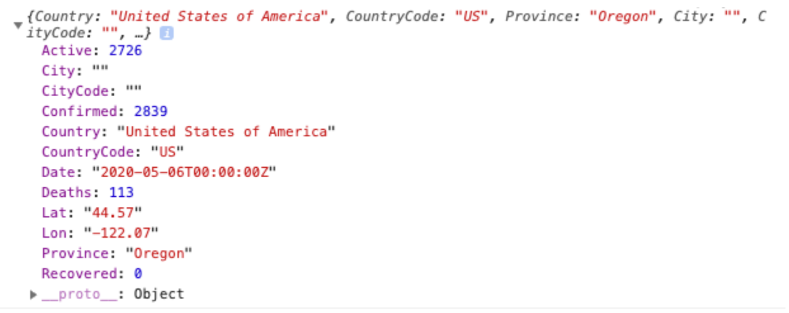
16. Refresh the page. Examine the console log and note that because we had nothing stored in local storage, we retrieved data from the API, saved the API response to local storage, set our locally stored **dateLastRetrieved** key value to today's date, then generated a random number to select a random object. Once we selected the object, we displayed its properties by updating the DOM elements on our newtab.html file.

You can expand the array of objects in the console to examine them by clicking on the arrow:



```
(57) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
{…}, {…}, {…}, {…}, {…}, {…}, {…}]
 ▶0: {Country: "United States of America", CountryCode: "US", Province: "Northern Maria…
 ▶1: {Country: "United States of America", CountryCode: "US", Province: "Hawaii", City:…
 ▶2: {Country: "United States of America", CountryCode: "US", Province: "Colorado", Cit…
 ▶3: {Country: "United States of America", CountryCode: "US", Province: "Arizona", City…
 ▶4: {Country: "United States of America", CountryCode: "US", Province: "Iowa", City: "…
 ▶5: {Country: "United States of America", CountryCode: "US", Province: "North Carolina…
 ▶6: {Country: "United States of America", CountryCode: "US", Province: "Wyoming", City…
```

You can also expand the randomly selected object to view its properties. Example:



```
▼{Country: "United States of America", CountryCode: "US", Province: "Oregon", City: "", C
 ityCode: "", …}
   Active: 2726
   City: ""
   CityCode: ""
   Confirmed: 2839
   Country: "United States of America"
   CountryCode: "US"
   Date: "2020-05-06T00:00:00Z"
   Deaths: 113
   Lat: "44.57"
   Lon: "-122.07"
   Province: "Oregon"
   Recovered: 0
 ▶__proto__: Object
```

17. Now that we've retrieved and stored data locally, the program acts a little differently.

    For the duration of today, each time a new tab is opened, the program will compare today's date to the **lastRetrievedDate** value in local storage and determine whether it's a new day and time to pull from the API again.

    Refresh the page. Note that the console log reflects that API data was already retrieved for today, so the program skips fetching API data and pulls the data from local storage, instead.

    Tomorrow, the program will fetch new data from the API. Hopefully, tomorrow's Covid-19 stats will look brighter.

    Stay safe and be well.