Sohee Cho
Week 6 Homework Part 2

1. What is the fetch Api?
    - The fetch API is a promise based API (application programming interface)
    - a. What does it replace the functionalities of?
        - It replaces the functionalities of the XMLHttpRequest, which is at the heart of what we know as AJAX
    - b. What does it provide an interface for?
        - It provides an interface for fetching resources from around the web, or even from our own projects.
2. What is a JavaScript Promise?
    - A JavaScript promise is an object that represents the eventual completion or failure of an asynchronous operation and its resulting value.
3. Why do developers like to use the fetch Api? And especially in conjunction with what? Give two reasons.
    - Developers like to use the fetch API because it retrieves data from a URL without a full page refresh
    - It is also more powerful than XHR (Ajax) and much more flexible
4. What is the fetch Api completely based on?

- It is based on promise, which is an object that represents the eventual completion or failure of an asynchronous operation.

5. What is the function declaration syntax of a regular named function?

    - function name(parameter) {

      [statement]

      }

6. What is the first thing the declaration of regular named function begins with?

    - The first thing the declaration of a regular named function beings with “function”

7. What is that first thing followed by?

    - It is then followed by the name of the function

8. What rules do function names follow?

    - Function names must follow the same rules as variables. They can contain letters, numbers (but not as the first character of the name), underscores, and $ signs.

9. What is promise chaining represented by in our Free IP Geolocation Api project?

    - Promise chaining is represented by the “.then ( )” blocks. It is indicated by the “.” which is before “then ( )”.

10. Why do we use arrow functions inside of the .then() blocks?

- We use arrow functions inside of the ".then()" blocks so that our objects can be seamlessly passed from one .then() block into another, without having to add a lot of extra code that is implicitly achieved with chaining.

11. What is the GoeLocation Api?

    - It is a web api that allows the user to provide their location to web applications if they want to. For example, if you are on "petco.com" and use "locate stores near me", the user will be prompted to allow permission for location.

12. What must applications add to their application code in order to be able to use the Geolocation Api? There are two pieces involved. Explain what each piece represents.

    - Applications must add the geolocation permission property "navigator.geolocation" to their code. The user's device will prompt the user to allow location access the first time the location is requested.

    - It returns a geolocation object that gives web content access to the location of any device. This allows a website or app to provide customized results based on the user's location.

13. How does the user allow the application to access their location? Use the application that you build to find out how.

    - The user allows the application to access their location by responding to a pop-up window which asks if the user allows or blocks access in their current browser.

    - For this project we use a condition to check whether or not the

Geolocation API is available in browsers. We use the “if/else” statement:

```
If (!navigator.geolocation){

        status.textContent = `Geolocation is not supported by your browser.`; }

Else {

        status.textContent = `Now locating...`;

        navigator.geolocation.getCurrentPosition(geoSuccess, error); }
```

14. If the Geolocation Api is available in the active browser window, what built-in Geolocation method do we use to get the location of the user by longitude and latitude coordinates?

    - We set the “.getCurrentPosition()” method on navigator.geolocation. It gets the user’s current location by longitude and latitude.

```
function geoSuccess(position) {

const latitude = position.coords.latitude;

const longitude = position.coords.longitude;

status.textContent = ``;

mapSource.href = `https://www.openstreetmap.org/#map=18/${latitude}/${longitude}`;
```

```
mapSource.textContent = `Your Coordinates are
Latitude: ${latitude} ° and Longitude: ${longitude} °`; }
```