Samuel Herscher

Homework #6

COMD3363 Dynamic Web

1. Fetch Api is a promise based api and it replaces the functionalities of the XMLHttpRequest. It provides an interface for fetching resources from around the web and from within your own file.
2. A JavaScript Promise is an object represents the soon to be completion or failure of an asynchronous operation and its end value.
3. Developers like to use fetch Api because it’s really simple to use. Especially in conjunction with promise chaining. 2 reasons are cause of its flexibility and because it’s so powerful.
4. Fetch Api is completely promised based.
5. Function syntax is the function key followed by then name of the function, regular named function is defined, or declared with the function keyword.
6. The first thing the declaration of a regular named function begins with Function.
7. Then it is followed by the name of the function.
8. Function names following the following rules; they contain letters but can’t be the first character and if there is more than one part in the name, they can have underscores and $ signs.
9. The promise chaining is represented by the “.then()” blocks, Indicated by the “.” That is before the “then()”.
10. We used arrow functions inside of the .then() blocks so our objects can pass seamlessly from one .then() to another without using a lot of extra code.
11. Geolocation Api is a web Api that allows the user to provide their location to web applications if they want to opt for it.
12. Applications that want to use this feature must add geolocation permission property “navigator.geolocation” to their code. Which will prompt the user’s device 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 devices. This allows a website or app to use customized results based on the users provided location.
13. The user allows the application to access their location by the user responding to a pop-up window that asks if they will allow or block access inside their current web browser. For the project we did we used condition to check if the Geolocation API is available and if it was used or not. The statement we used 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);}

1. If the geolocation is available in the active web browser we will use the Geolocation API built in method called .getCurrentPosition().