When we click the 'update button', the function name 'GetWeatherInfo' works.

Based on what options we choose, in the function, it will get value for what we chose int the option.

After getting value of the option, it will create 'url' variable of the city

According to the code, it used the format like this:

api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}

After that, there are two ways to present the current weather.

Basically using XMLHttpRequest and fetch are used to interact with server.

1.using XHR

--> when we use XHR, we create an object from XMLHttpRequest.

--> then we set the response type to 'json'

--> use 'onreadystatechange' function. There are 0 to 4 states for server request.

when the states for server request is finished and ready to response, which is 4,

and if the status code is 200, which means it works fine, then it gets json.

With the json, it uses the data from json to represent the weather status.

In this code, XHR is used asynchronous.

When the value of readystate is 0, request is not yet initialized.

When the value of readystate is 1, open() has been called.

(when open is called, it initialize the request. we can set the method type 'get' or 'post'. Also, we put 'url' and put the boolean value true at the last parameter to make it asynchronous.)

syntax --> open(method, url, ‘true’) --> method is 'get' or 'post'

After the open() is called, the ready state is 2, which represents that send() has been called.

(send() is basically used to send our request to server. Since we are using asynchronous, it returns the data from server right away after we give request to server.)

when ready state is 3, it means it is loading.

when it is 4, it recieves the result that we have requested and the data is ready to use. We can check it if it is successful or failure by checking the status code.

After that, this function uses data from json and shows the weather status based on the json.

2. using fetch

syntax --> fetch('url').then().then()

1.In fetch(), the function gets url as an argument in the parameter.

Basically fetch works as 'get' method as a default and returns Promise Object.

.then() is used as chaining method.

2. In the code, it uses .then() method after fetch..

The first 'then()' gets ‘Response object’ as an argument and works when the operation is completed successfully and returns 'response.json()'

3. After that, it uses .then() methd after .then()

Firstly it gets an argument 'data' in the parameter.

It calls the function 'ProcessJSON(data)' by passing an argument 'data' as parameter.

The 'data' is the response.json() which was returned from first '.then()'

After the first '.then()' is working successfully, it calls the function 'ProcessJSON()'

If there are any errors, .catch() method works right away and shows alert message.

After the data is processed as json, the data is used to present the current weather information.