Map Method:

Step 1: Have defined variable – var array = [#, #, #, #];

Step 2: Define new variable for mapped array to equal – var newArray = array.map(array => ((function action)) );

Step 3: Display the new variable in the console - console.log(newArray);

Key points 🡪 - Does not need to be a numerical operation

- Must perform a function in the map parameter

- Acts similar to a for loop but in more concise code form

Filter Method:

Step 1: Have defined variable – var array = [#,#,#,#,#,#];

Step 2: Define new variable to hold the newly filtered array – var filteredArray = array.filter(array => ((function action)) );

Step 3: Display the variable in the console – console.log(filteredArray);

Key points 🡪 - Filter can return an array that has a different length than the input array’s length

-

Sort Method:

Slice Method:

Step 1: Have defined variable – var array = [#,#,#,#,#,#];

Step 2: Define new variable for sliced data values from previous array – var slice1 = array.slice(0, #);

Key points 🡪 - Numbers in the slice parameters indicate index points, and can grab any values within the range of the index points.

- The values in the parameter must be treated as a range and will return the values as such with (startPoint, endPoint)

Leaflet GeoJSON Functions:

**onEachFeature Function:**

L.geoJSON(data, {

onEachFeature: function(feature, layer) {

layer.bindPopup();

}

});

**pointToLayer Function:**

L.geoJSON(data, {

pointToLayer: function(feature, latlng) {

return L.marker(latlng);

}

});