JavaScript Fetch:

JavaScript fetch is a way to access data asynchronously so that the computer doesn’t have to wait for a response before executing other tasks. It returns a promise that resolves with the response. It is mainly used for APIs. A promise is an object that represents the completion or failure of an asynchronous operation, meaning it is what you get when you make an async call that will eventually resolve and gives the response. You can use promises without fetch by having any type of asynchronous code. This could be with setTimeout, with AJAX, which is an alternative to fetch, or with Axios, which is another alternative to fetch. You can use fetch to receive a json file by using await and then if the response is okay you get the response.json which will return the json from the response. You can use fetch to retrieve data from a REST API by simply using fetch with the API’s url like “fetch(url)”.

APIs (Application Programming Interface):

An API is a way for a user to access data that is being provided through a software using different HTTP methods, like GET, POST, PUT, and DELETE. You could use the national park service API to get information by sending a query of the information you want through a fetch, then using javascript to parse and render the information that you receive. To get Yellowstone you would target the parks link with the keyword parkCode=yell which represents Yellowstone. To get all parks in a state you would include a stateCode.

Coding as a team:

Some common challenges that arise for inexperienced programmers writing code as a team are coordinating schedules, equal contribution, unclear roles, and staying on the same page.