The Dollar ($) Identifier

The dollar sign is commonly used as a shortcut to the function *document.getElementById()*.

This is mostly used in jquery

Splice function in JavaScript

The splice() changes the contenet of the array by removing, adding or replacing array elements.

It takes three arguments: splice ( <start>, <delete count> , <new elements>) .

Start: this is the **start index** from where changes has to start.

Delete count: It indicates the number of elements to be removed from **start index** of the array.

If its value is 0 or –ve then no elements are removed, also you need to specify at least one new element.

New element: element to be added to the array beginning from the **start index** of the array.

**Difference between axios, fetch and http ?**

|  |  |
| --- | --- |
| **Axios** | **Fetch** |
| Axios is a stand-alone 3rd party package that can be easily installed into a React project using NPM. [ npm install axios ] | Unlike Axios, fetch() is built into most modern browsers. With fetch you do not need to install a third party package. |
| **So its up to you, you can go with fetch() and potentially mess up if you don't know what you are doing OR just use Axios which is more straightforward in my opinion.** | |
| Axios just returns the data object you would expect. | If you use .fetch() there is a two-step process when handing JSON data. The first is to make the actual request and then the second is to call the .json() method on the response. |
| axios.get(url).then(response => console.log(response)); | fetch(url).then(response => response.json()).then(data => console.log(data)); |
| **Handling error responses** | |
| How does Axios handle this? The way you would probably expect. You get any kind of error with the http request and the .catch() block is executed. | Logically you would think that if .fetch() gets an error it would enter the .catch() block and return anything there, right? Not necessarily. |
| get(url).catch(error => console.log('BAD', error)).then(response => response.json()); | fetch(url).catch(error => console.log('BAD', error)).then(response => response.json()); |

**Await keyword?**

The await operator is used to wait for a [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise). It can only be used inside an [async function](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function).

The await expression causes async function execution to pause until a Promise is settled, that is fulfilled or rejected, and to resume execution of the async function after fulfillment. When resumed, the value of the await expression is that of the fulfilled Promise.