1. Select **five methods** that can be used on an Array and describe the following for each:

1) what the method signature is

2) what the method does

3) why would this method be useful (how could you use it)?

Method 1- Push will add element/s to the array. When you push elements it will add on to the end of the array. This is helpful when you are needing to add additional elements later on in the code. You can just take the array name and push the new elements to it.

Method 2- Pop will remove a single element from an array. It will removed the very last element, returning the removed element and changing the array. This can be used if you added the wrong elements to the wrong array. An example is when you added a tomato to the scheduling array and you meant to add it to the salad array. You can just remove it by scheduling.pop(); and it will remove the tomato that you added by mistake.

Method 3- Join will allow you return all the elements in a string. This is useful when you are trying to return a large subset of data. You can join 2 different arrays to make a new one.

Method 4- Sort method converts the elements into a string and then sorts them in ascending order defaulting. This method is used widely in coding. This is great when you are trying to sort your data and put it in order.

Method 5- Reverse will allow you to reverse the elements' positions in the array. This is will be useful when you are trying to reverse the order of the data. You could even combine the sort array with the reverse if you are looking for a specific order for the data to be in.

https://www.freecodecamp.org/news/the-javascript-array-handbook/

2. What is the difference between == and ===?

The ==. This operator will follow loose equality comparison algorithm meaning it will perform the type conversion of one operand in order to make the data types of the operands the same. The == returns true if operand have the same data type and same values, returned false if the values differ.

The === is a strict equality. This operator will follow strictly equality comparison algorithm meaning it will not do the type conversion before comparing their value and return false if the data is not the same. Using === will return true only if operands are of the same data type and same value, otherwise returns false. Since both operands are used to compare two values as equals, it is important when writing code that you have the correct operator or it will cause errors in your code.

https://www.scaler.com/topics/javascript/difference-between-double-equals-and-triple-equals-in-javascript/

4. What is your favorite thing you learned this week? I really like learning about adding from each week. I like being able to take a larger subset of data and be able to set I really struggled with the more complex scenarios and adding in all the ifs and else ifs.