Array Manipulation Activity

#### **Learning Outcome Addressed**

  1. Manipulate arrays and strings in JavaScript

Follow the instructions in Next Tech to complete the activity.

**Slicing**

### Array and String Manipulation in JavaScript

JavaScript provides built-in methods and functions that will help you a lot when programming. These functions can especially be helpful when dealing with strings and arrays.

Let's explore the following built-in methods for arrays and strings in JavaScript: slice(), splice() and split().

**slice():** copies a given section of an array, and returns a new array containing that copy. **It does not change the original array**

GivenArray.slice(from, until)

from: Slice the GivenArray **starting** at this element index.

until: Slice the GivenArray **up to but not including** this element index.

For example, lets consider the following array:

arr = ["A","B","C","D"]

To slice the first three elements in this array, the from parameter would be 0 (the first element in an array is always indexed at 0).  
The until parameter needs to be 3. Note that the slice() method does not include the last given element.

array[0] --> A // included

array[1] --> B // included

array[2] --> C // included

array[3] --> D // not included

### Using slice()

**Your task in this activity is to write a function called sliceElements() that will return the last three elements of any given array. The array can be of any length, as long as it is at least three elements.**

Task

* Write a function called sliceElements that return the last 3 elements of any given array

### Array Manipulation in JavaScript

**splice():** The naming of this function is similar to slice() and often causes confusion for developers. The splice() method **changes the original array** by removing or replacing elements from it.

**Removing elements:**

Splice takes at least one argument, but can include more optional arguments:

* The first argument is the starting index where you want to start removing items.
* The second argument is the number of elements you want to remove
* The third optional argument is any new elements that would replace other elements in the array. Splice returns an array with the removed elements.

GivenArray.splice(index, number of elements to remove, optional elements to insert)

* index: The index is the starting point of where to remove elements.
* number of elements: This is the number of elements to remove. If left empty, every element starting from the given index (the first argument) will be removed from the array.
* optional elements to insert into the array: When defined, these elements will be inserted into the array at the specified index in the second argument.

For example, consider the following array:

const numbers = [1, 2, 3, 4, 5];

If you want to **remove** one element starting at the second index of the array, you would call:

numbers.splice(2, 1);

Since splice() returns an array containing the elements you've removed, you would get a return value of:

// Expected result -> [3]

Now, if you call:

console.log(numbers);

You will see that the numbers array has had one element removed from it at its second position.

//Expected result -> [1, 2, 4, 5];

Compare this to the original numbers array. Can you see which number was spliced out?

Here is another example where **every element starting at a certain index is removed.**

const arr = ["A","B","C","D"]

Calling splice() on const arr will result in the following:

arr.splice(2); // Every element starting from index 2, will be removed

//Expected result -> ["A", "B"]

arr.splice(2,1); // 1 element starting from index 2, will be removed

//Expected result -> ["A", "B", "D"]

This example inserts something into an array. In this example, 'h' and 'i' are inserted at the first index, and nothing is removed.

let arr = [1,2,3,4,5];

arr.splice(1, 0, "h", "i");

arr

//Expected result ->[1, "h", "i", 2, 3, 4, 5]

Now for a tricky one. This inserts "h" and "i" at the third index, and removes everything in the original array from the second index on.

let arr = [1, 2, 3, 4, 5];

arr.splice(3, 2, "h", "i" )

arr

//Expected result ->[1, 2, 3, "h", "i"]

**Adding elements:**

You can also add some letters to the array without removing any other elements. To do so, set your count (the second argument) to zero. With splice() you can add elements to an array at any point in the array, not just at the end, as you learned with array.push(). In this example, "x" and "y" will be added into the middle of the letters in the const letters array, at the second index.

const letters = ["a", "b", "c", "d"];

letters.splice(2, 0, "x", "y");

console.log(letters);

//Expected result -> ["a", "b", "x", "y", "c", "d"]

**Replacing elements:**

To replace array elements using the splice() method, you can simply pass the index number of where you want to start replacing elements along with any new elements as additional parameters to the method:

array.splice(index to start replacing, number of elements to replace, elementToAdd1, elementToAdd2, ...elementToAddN)

This example shows how to replace one element at the second index of the letters array with another letter.

const letters = ["e", "f", "g", "h"]

letters.splice(2, 1, "x");

console.log(letters);

//Expected result -> ["e", "f", "x", "h"]

You can also replace elements with more than one element by adding extra arguments with the elements you want to insert. This example will show how to replace two elements in the letters array with three additional letters, starting at index two.

const letters = ["a", "b", "c", "d", "e", "f"];

letters.splice(2, 2, "x", "y", "z");

console.log(letters);

//Expected result -> ["a", "b", "x", "y", "z", "e", "f"];

As you can see, two elements were removed from the array ("c" and "d"), and three letters were put in at index two ("x" "y" and "z").

Don't worry if this is somewhat tricky right now. Splice() is a bit of a tricky built-in method, but with practice you will soon get the hang of it!

### Using splice()

**Your task in this activity is to write a function called spliceElements that will remove the last element of a given array and add two new elements at that position.**

### Array Manipulation in JavaScript

**split():** While splice() and slice() are used for arrays, split() is used for strings. It divides a string into substrings and returns them in an array.

givenString.split(separator, limit)

* separator: Describes **how to split** a string, by defining on what to split the string. This can be done, for example, on every comma, space, special character.
* limit: Defines the number of splits.

Both of these arguments are optional. Let’s see how this works in the example below. You'll notice:

* (str.split(' ')) splits const str on every empty space
* (str.split('')) splits const str on every character ,
* (str.split(' ', 3)) splits const str on every space AND returns only the first three elements in the resulting array .

const str = 'MIT Certificate loves JavaScript';

const words = str.split(' ');

console.log(words[3]);

// expected output: "JavaScript"

const chars = str.split('');

console.log(chars[0]);

// expected output: "M"

const splits = str.split(' ', 3)

console.log(splits)

// expected output: Array ["MIT", "Certificate", "loves"]

### Using split()

**Your task in this activity is to write a function called splitElements that will convert a given string into an array of words**