**Array Methods by building a Music Player**

**Algorithm**

- step 1: create a variables and DOM through its ID and a audio HTML element

- step 2: Create a variable "allSongs" array of objects 10 having all song details

- step 3: create a Object assigned with allsongs using spread operator

- step 4: create an arrow function "renderSongs" with map() creating element and button element

- step 5: Assign songsHTML to the playlistSongs to display

- step 6: call renderSongs function with userData?.songs

- step 7: create a variable called sortSongs sort the songs from userData?.songs.sort()

**IMPORTANT NOTES**

- The **Web Audio API** and how to use it to play songs. All modern browsers support the Web Audio API, which lets you generate and process audio in web applications.

- The **spread operator (...)** allows you to copy all elements from one array into another. It can also be used to concatenate multiple arrays into one. In the example below, both arr1 and arr2 have been spread into combinedArr:

- An **arrow function** is an anonymous function expression and a shorter way to write functions. Anonymous means that the function does not have a name. Arrow functions are always anonymous.

- To create a named arrow function, you can assign the function to a variable

- IMPLICIT RETURN in Arrow Function

If the arrow function is returning a simple expression, you can omit the return keyword and the curly braces {}. This is called an implicit return.

- **Map()** Method : The map() method is used to iterate through an array and return a new array. It's helpful when you want to create a new array based on the values of an existing array

Notice that the map() method takes a function as an argument. This is called a callback function, which is a function that is passed to another function as an argument.

- The **join() method** is used to concatenate all the elements of an array into a single string. It takes an optional parameter called a separator which is used to separate each element of the array.

- **Optional chaining (?.)** helps prevent errors when accessing nested properties that might be null or undefined.

ex: const zipCode = user.address?.zipCode;

// Returns undefined instead of throwing an error as zipCode is not available in the object

- **The sort()** method converts elements of an array into strings and sorts them in place based on their values in the UTF-16 encoding.

- Strings are compared **lexicographically** which means they are compared character by character. For example, "Apples" is less than "Bananas" because "A" comes before "B" in the alphabet.

- The reason why this example is returning numbers is because the sort() method is expecting a number to be returned. If you return a negative number, the first item is sorted before the second item

- It should return a number where: - A negative value indicates that a should come before b. - A positive value indicates that a should come after b. - Zero or NaN indicates that a and b are considered equal.

To memorize this, remember that (a, b) => a - b sorts numbers in ascending order.

If omitted, the array elements are converted to strings, then sorted according to each character's Unicode code point value.

- The find() method retrieves the first element within an array that fulfills the conditions specified in the provided callback function. If no element satisfies the condition, the method returns undefined.

- ?. -- optional chaining operator

- play() is a method from the web audio API for playing an mp3 file.

- pause() is a method of the Web Audio API for pausing music files.

- The indexOf() array method returns the first index at which a given element can be found in the array, or -1 if the element is not present.

- The forEach method is used to loop through an array and perform a function on each element of the array.

- The filter method keeps only the elements of an array that satisfy the callback function passed to it:

- createElement() is a DOM method you can use to dynamically create an element using JavaScript. To use createElement(), you call it, then pass in the tag name as a string:

- The createTextNode() method is used to create a text node. To use it, you call it and pass in the text as a string:

- element.id would set an id attribute, and element.ariaLabel would set an aria-label attribute. Both of them accept their values as a string

- appendChild() lets you add a node or an element as the child of another element.

- The "**ended**" event listener is appropriate for this. It is fired when the playback of a media reaches the end.

**Palindrome or Not**

**Steps:**

1. You should have an input element with an id of "**text-input**".
2. You should have a button element with an id of "check-btn".
3. You should have a div, span or p element with an id of "result".
4. When you click on the #check-btn element without entering a value into the #text-input element, an alert should appear with the text **Please input a value.**
5. When the #text-input element only contains the **letter A** and the #check-btn element is clicked, the #result element should contain the text **A is a palindrome**. -- if its only(single) letter
6. When the #text-input element contains the text **eye** and the #check-btn element is clicked, the #result element should contain the text eye is a **palindrome**.
7. When the #text-input element contains the text **\_eye** and the #check-btn element is clicked, the #result element should contain the text **\_eye is a palindrome.**
8. When the #text-input element contains the text **race car** and the #check-btn element is clicked, the #result element should contain the text **race car is a palindrome**.
9. When the #text-input element contains the text **not a palindrome** and the #check-btn element is clicked, the #result element should contain the text not a palindrome is **not a palindrome.**
10. When the #text-input element contains the text **A man**, a plan, a canal. Panama and the #check-btn element is clicked, the #result element should contain the text A man, a plan, a canal. Panama is a palindrome.
11. When the #text-input element contains the text **never odd or even** and the #check-btn element is clicked, the #result element should contain the text never odd or even is a palindrome.
12. When the #text-input element contains the **text nope** and the #check-btn element is clicked, the #result element should contain the text **nope is not a palindrome**.
13. When the #text-input element contains the text **almostomla** and the #check-btn element is clicked, the #result element should contain the text **almostomla is not a palindrome.**
14. When the #text-input element contains the text **My age is 0, 0 si ega ym.** and the #check-btn element is clicked, the #result element should contain the text My age is 0, 0 si ega ym. is a palindrome.
15. When the #text-input element contains the text **1 eye for of 1 eye**. and the #check-btn element is clicked, the #result element should contain the text **1 eye for of 1 eye. is not a palindrome.**
16. When the #text-input element contains the text 0\_0 (: /-\ :) 0-0 and the #check-btn element is clicked, the #result element should contain the text **0\_0 (: /-\ :) 0-0 is a palindrome**.
17. When the #text-input element contains the text **five|\\_/|fou**r and the #check-btn element is clicked, the #result element should contain the **text five|\\_/|four is not a palindrome**.