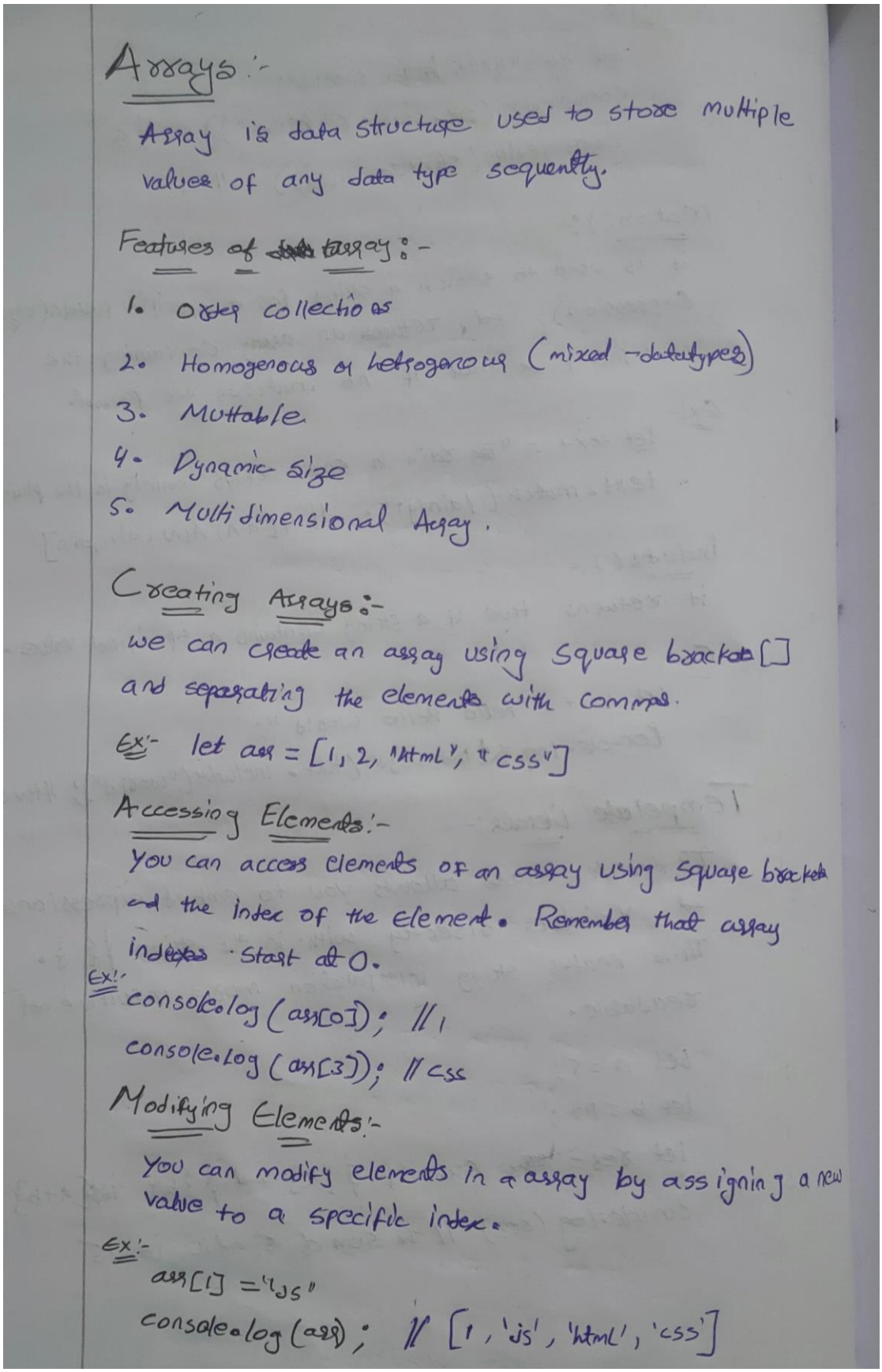
Array Methods:



Arrays :-

Array is data structure used to store multiple values of any data type sequently.

Features of array :-

1. Ordered collections
2. Homogenous or heterogenous (mixed - datatypes)
3. Muttable
4. Dynamic size
5. Multi dimensional Array.

Creating Arrays :-

we can create an array using square brackets [] and separating the elements with commas.

Ex:- let arr = [1, 2, "html", "css"]

Accessing Elements :-

You can access elements of an array using square brackets and the index of the element. Remember that array index start at 0.

Ex:-

```
console.log(arr[0]); // 1
console.log(arr[3]); // css
```

Modifying Elements :-

You can modify elements in a array by assigning a new value to a specific index.

Ex:-

```
arr[1] = "js"
console.log(arr); // [1, 'js', 'html', 'css']
```

dynamic size :-

we can increase array size dynamically

Ex:- var arr = ["hello", "world", 2];
arr[3] = 3;
console.log(arr); //["hello", "world", 2, 3]

Multidimensional Array :-

Ex:- var arr = [1, 2, 3, [4, 5, [6, 7, 8]]];
console.log(arr[3][2][2]);

Array Methods :-

Java script provides many built in methods to work with arrays such as push pop, shift, unshift, slice, splice, concat, indexOf, includes, & many more

1. Array length :-

length property returns the number of elements in an Array

Ex:-
let arr = [1, 2, "html", "css"]
console.log(arr.length); // 4

2. Array at() :-

The at() method of Array instance takes an integer value and returns the item at the index, allowing for positive and negative integers. Negative integers count back from the last item of an array

* Print last index value of array (length-1)

Result : console.log(arr[arr.length-1]);

Example :-

let arr = [1, 2, 3, 4];
console.log(arr.at(0)); // 1
console.log(arr.at(-1)); // 4

3. Concat():- Method is used to merge two or more arrays. It does not modify the existing arrays but instead returns a new array containing the elements of orginal arrays consatenated together.

Ex:-

```
var arr1 = [1, 2, 3]
var arr2 = ['a', 'b', 'b']
console.log(arr1.concat(arr2)); //[1,2,3,'a','b','c']
```

• Loops in arrays:-

```
var a = [1, 2, 3, 4, 5, 6, 7, 8]
for (i=0; i<arr.length; i++){
    console.log(arr[i]);
}
```

```
for (i in arr) {
    console.log[i];
}
```

4. Array Splice():-

Splice() method is used to change the contents of an array by removing or replacing existing elements and/or adding new elements in place.

• It modifies the orginal array and return an array containing the removed elements.

Syntax:-

```
array.splice(startIndex, endIndex, add item 1, add item2);
```

Example:- // adding at specific index.

```
var arr = ["html", "css", "JS", "react"]
arr.splice(1, 0, "bootstrap");
console.log(arr); //[html, bootstrap, css, JS, react]
```

```
// replacing at a specific index
ass.splice(1,1,"bootstap");
console.log(ass); // [html, bootstap, JS, react]
// replacing all values at a time
ass.splice(0,4, "Java", "Python", "NodeJs", "mySql");
console.log(ass); [Java, Python, nodejs, mySql];
```

5. Assay Sliced:-

slice() method is used to extact section of an element assay and returns a new assay containing the extracted elements.

Syntax:- assay.slice(startIndex, endIndex);

Example:-

```
var ass =["html", "css", "js", "react"]
var ass2 = ass.slice(0,2);
console.log(ass2); // [html, css]

var ass2 =ass.slice(-1);
console.log(ass2); // [react]
```

6. Assay Pop():-

Pop method removes the last elemenact from an assay and returns the element.

7. Assay Push():-

Push method adds one ore more elements to the end of an assay and returns new length of assay.

Example:-

```
var ass = ['html', 'css', 'JS', 'react'];
ass.POP(); // ['html, css, JS];
ass.push("git");
console.log(ass); // [html, css, JS, git]
```

8. Array shift();

shift method() is used to remove the first element of an array and returns that element.

9. Array unshift();

unshift method adds one or more elements to the beginning of an array and return new length of an array.

Example:-

```
var arr = ["html", "css", "JS", "React"];
arr.shift(); // [css, JS, React]
arr.unshift("git"); // ["git", "html", "css", "JS", "React"]
```

10. Array Sort and reverse Methods:-

```
arr.sort();
arr.reverse();
```

Examples:-

```
var arr = [9, 3, 4, 67];
console.log(arr.sort());
```

Output:- 3, 4, 67, 9 [Lexographical order it takes only first letter of an element]

reverse Example:-

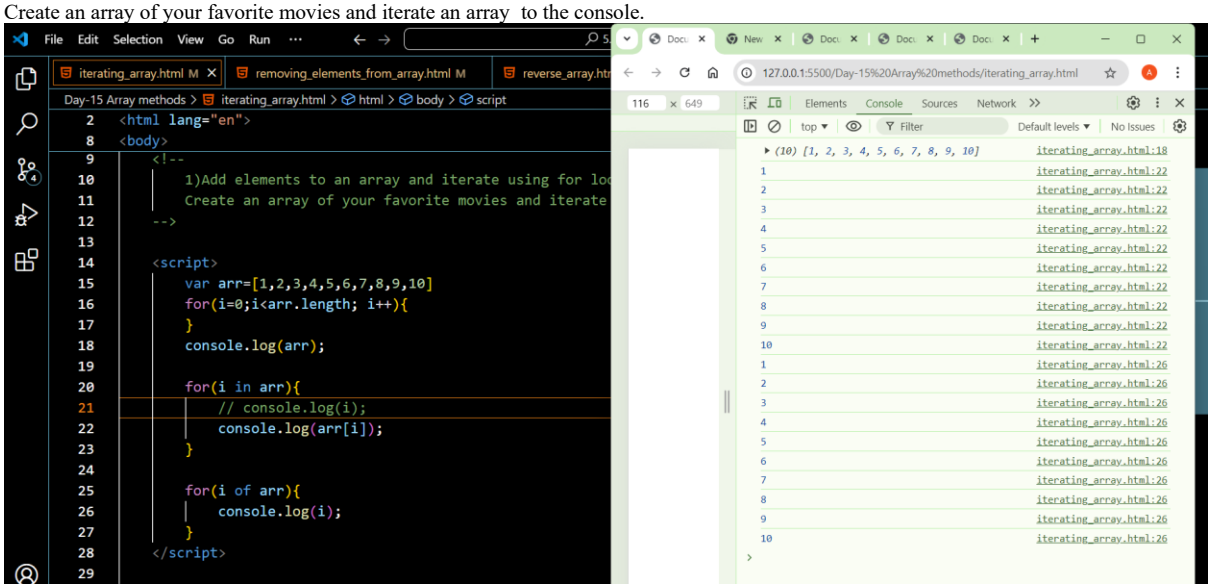
```
var arr = ["html", "css", "JS"];
console.log(arr.reverse()); // [JS, css, html]
```

Problem:- reverse an array and push into the new array

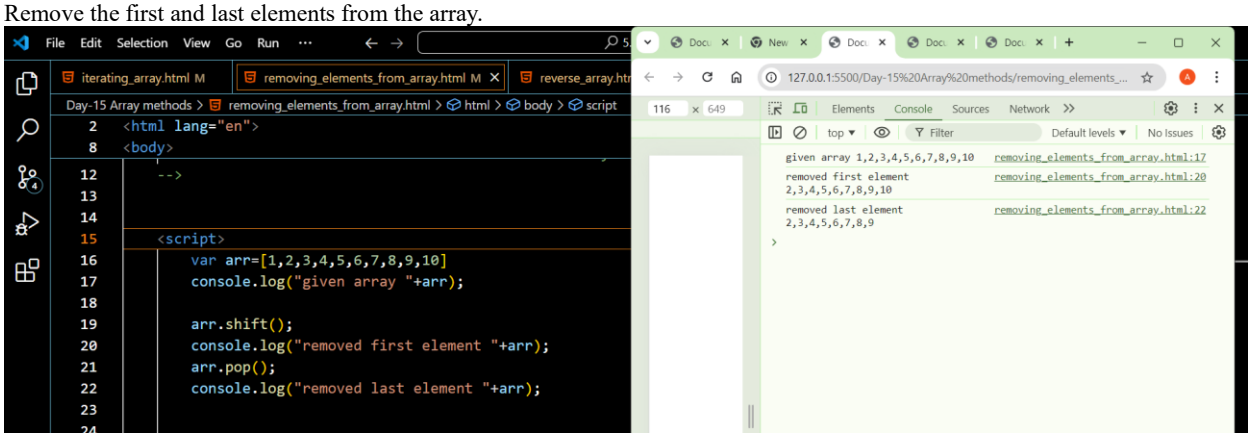
```
var a = ["html", "css", "JS", "React"]
var new = []
for(i=a.length-1; i>=0; i--){
    console.log(a[i]);
    new.push(a[i]);
}
console.log(new); // [React, JS, CSS, html]
```

TASKS:

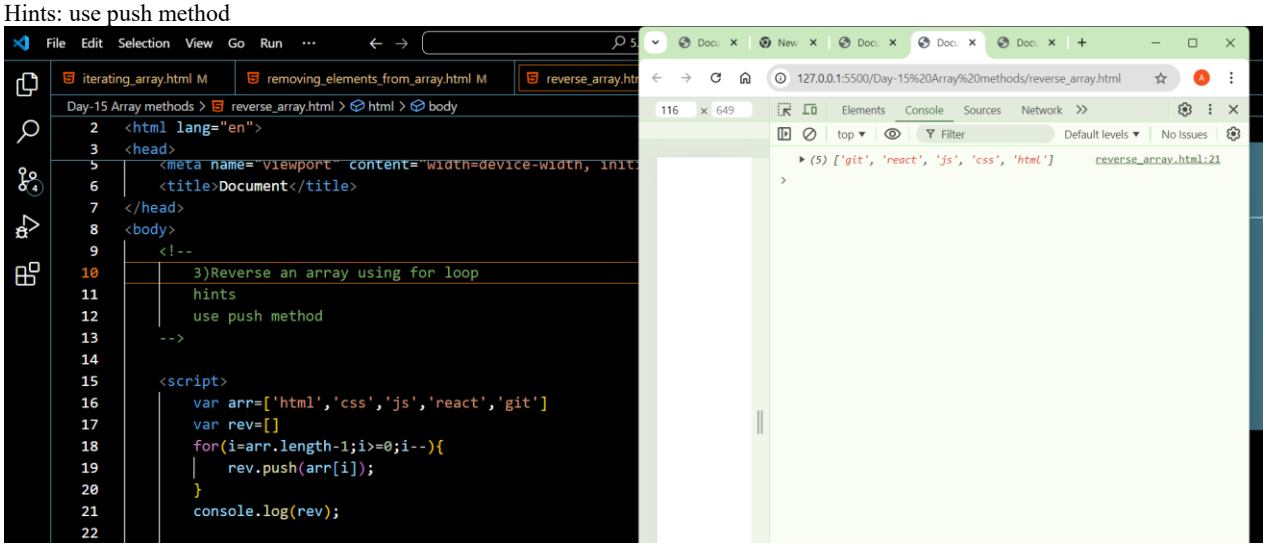
1)Add elements to an array and iterate using for loop and for in loop and for of loop
Create an array of your favorite movies and iterate an array to the console.



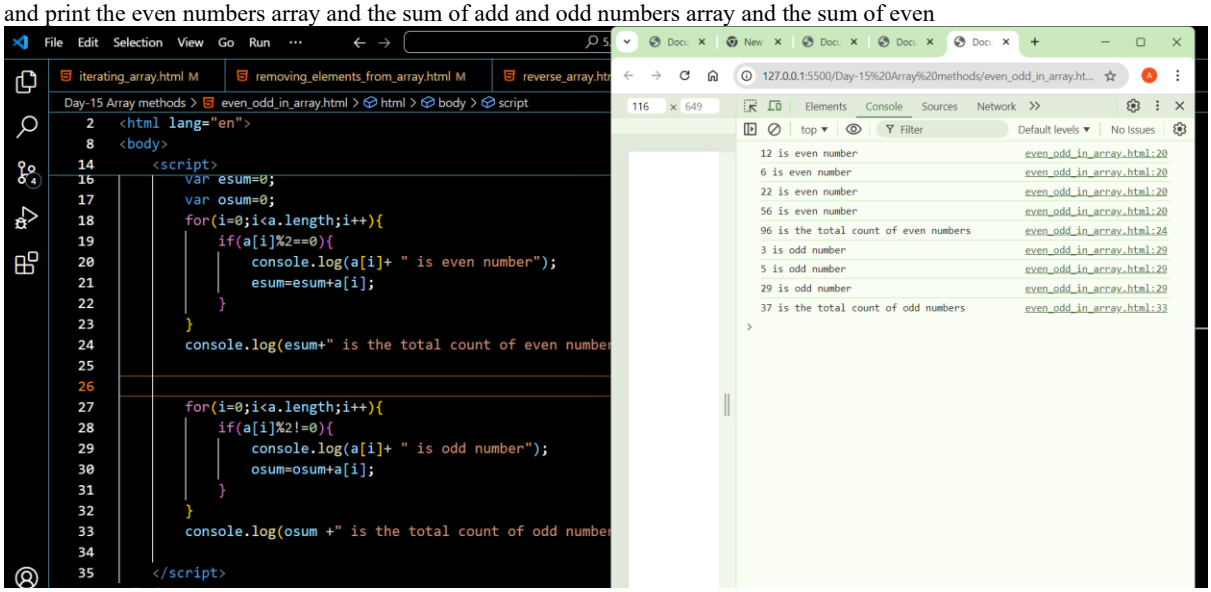
2)Remove elements from an array
Remove the first and last elements from the array.



3)Reverse an array using for loop
Hints: use push method



4) find the even and odd numbers in an array [12,3,5,6,22,56,29]
and print the even numbers array and the sum of add and odd numbers array and the sum of even



5) Take a heterogeneous array and separate each data type into new array
hints : use loop, typeof and push method
inp: let arr = ["apple", "banana", "mango", "banana", 3, 4, 5, 6, true, {name: "object"}];
out :
num=[3,4,5,6]
str=["apple","banana","mango","banana"]
bool=[true]
obj=[{name: "object"}]