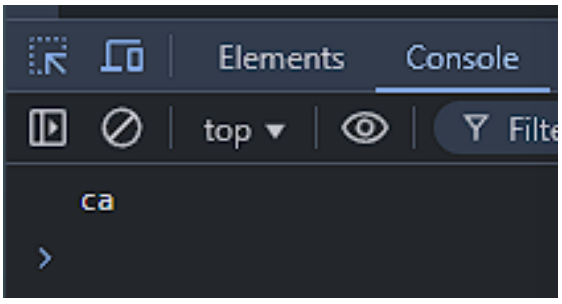
Day 4 - Class Assignment
14th Feb, 2025

- Remove all adjacent duplicates from a string:- https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string

Input ➖

```
var removeDuplicates = function(s) {
    let x = [];

    for (let i of s) {
        if (x.length > 0 && x[x.length - 1] === i) {
            x.pop();
        } else {
            x.push(i);
        }
    }

    return x.join('');
};

const s = "abbaca";
console.log(removeDuplicates(s));
```

Output ➖

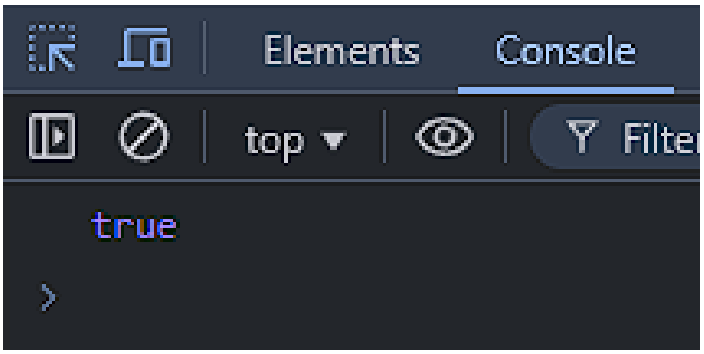


- Valid Anagram:- https://leetcode.com/problems/valid-anagram

Input ➖

```
var isAnagram = function(s, t) {
    // check length
    if (s.length !== t.length) return false;

    let map = new Map();

    // Count frequency of characters in string s
    for (let char of s) {
        map.set(char, (map.get(char) || 0) + 1);
    }

    // Check if characters in string t match the frequency counts in map
    for (let char of t) {

        if (!map.has(char)) return false; // t has a char not in s
        map.set(char, map.get(char) - 1); // Decrease the count
        if (map.get(char) < 0) return false; // More occurrences in t than in s

    }

    return true;
};

const s = "anagram";
const t = "nagaram";
console.log(isAnagram(s, t)); // Output: true
```

Output ➖

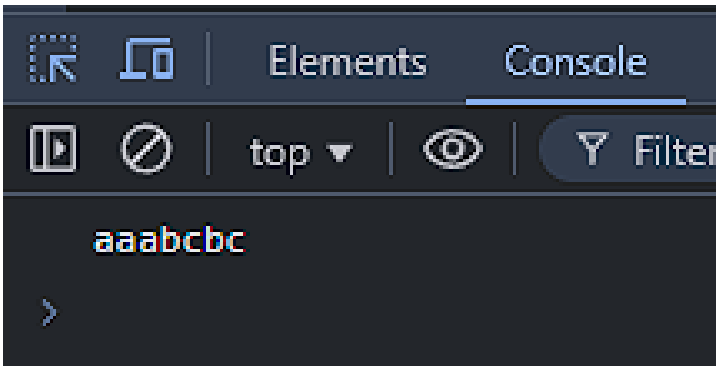


- Decode String:- https://leetcode.com/problems/decode-string/

Input

```javascript
var decodeString = function(s) {
    let stack = [];
    let currentNum = 0;
    let currentStr = "";

    for (let char of s) {
        if (!isNaN(char)) {
            // Build multi-digit number
            currentNum = currentNum * 10 + parseInt(char);
        } else if (char === "[") {
            // Push current number and string onto stack
            stack.push(currentStr);
            stack.push(currentNum);
            currentStr = "";
            currentNum = 0;
        } else if (char === "]") {
            // Pop number and previous string from stack
            let num = stack.pop();
            let prevStr = stack.pop();
            currentStr = prevStr + currentStr.repeat(num);
        } else {
            // Append character to current string
            currentStr += char;
        }
    }

    console.log(currentStr);
};

decodeString("3[a]2[bc]");
```

Output



- Reorganize String:- https://leetcode.com/problems/reorganize-string/

Input

```
var reorganizeString = function(s) {
    let freqMap = new Map();

    // Count frequency of each character
    for (let char of s) {
        freqMap.set(char, (freqMap.get(char) || 0) + 1);
    }

    // Sort characters by frequency (max heap simulation using array sorting)
    let maxHeap = [...freqMap.entries()].sort((a, b) => b[1] - a[1]);

    let maxFreq = maxHeap[0][1];
    if (maxFreq > Math.ceil(s.length / 2)) return "";  // Impossible case

    let res = new Array(s.length);
    let index = 0;

    // Fill characters in order of frequency
    for (let [char, freq] of maxHeap) {
        for (let i = 0; i < freq; i++) {
            if (index >= s.length) index = 1;  // Move to odd indices if even indices are full
            res[index] = char;
            index += 2;
        }
    }

    console.log(res.join(''));
};

reorganizeString("aab");
```

Output