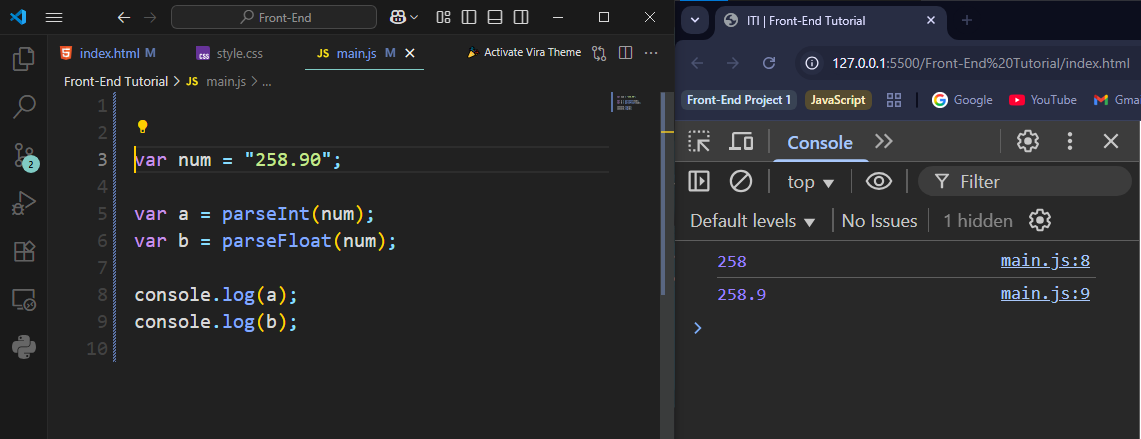
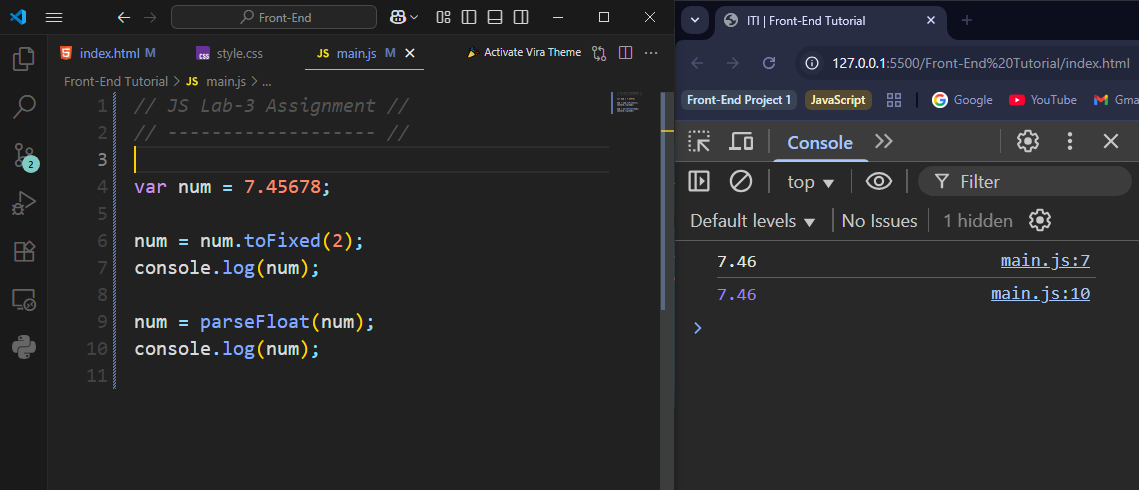
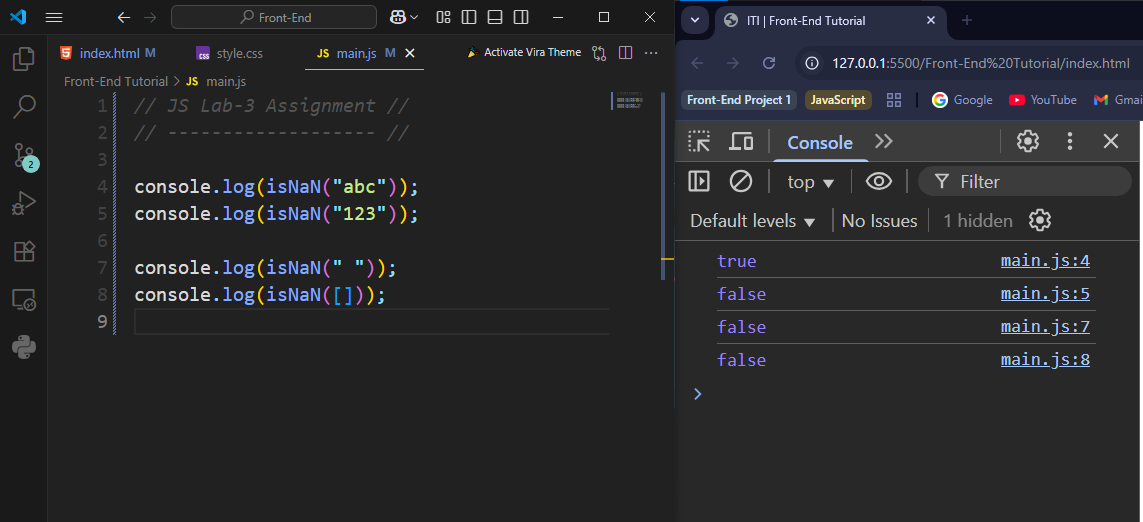
1. Convert the string "258.90" to: (a) integer, (b) floating number. Store in two variables.



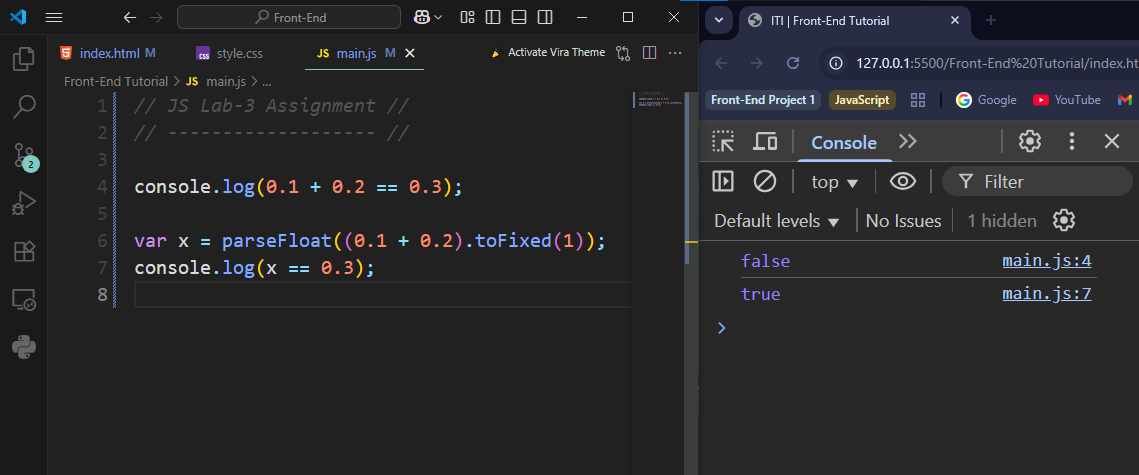
2. Format the number 7.45678 to exactly 2 decimal places (string) then convert it back to a number.



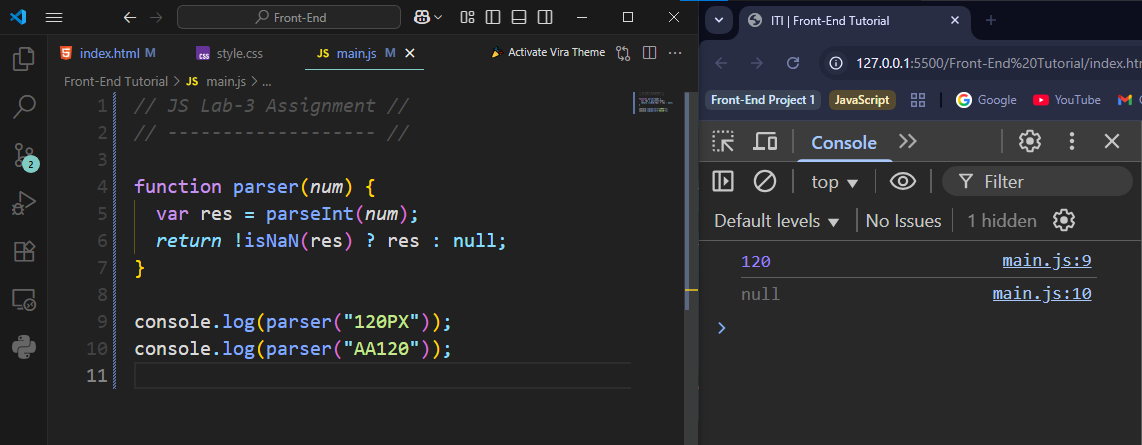
3. Check if the value 'abc' is NaN. Also show a case where isNaN returns false for a non-number.



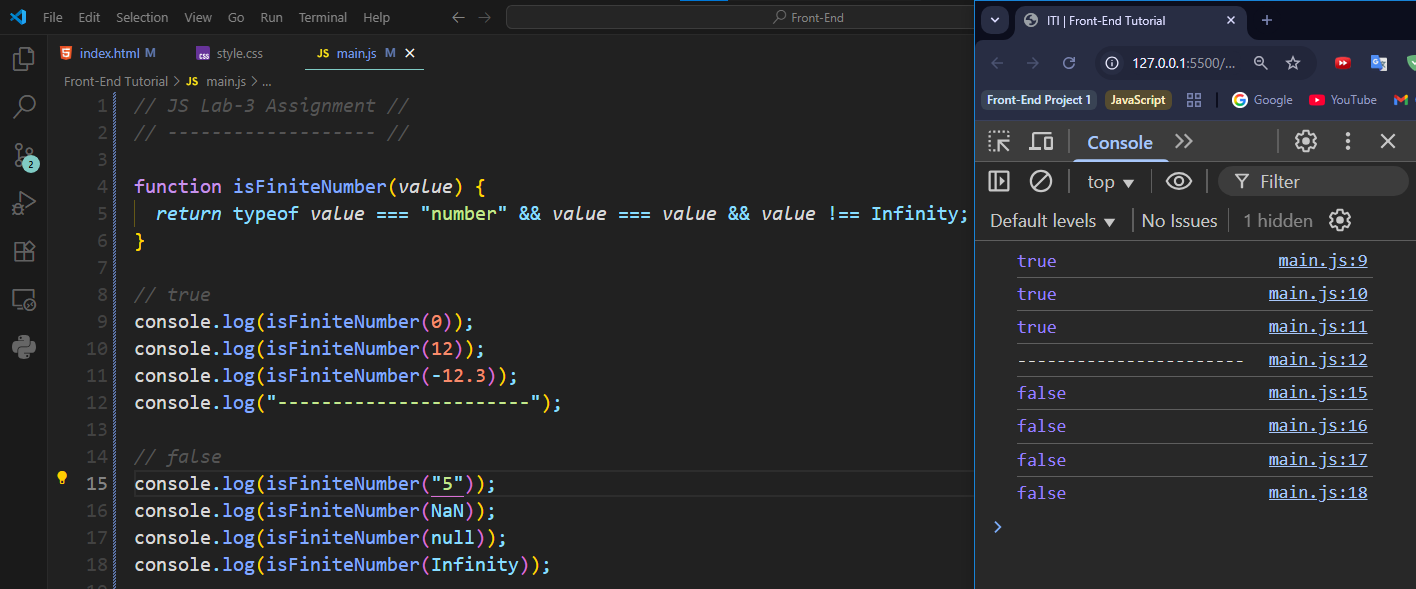
4. Floating point: Show that (0.1 + 0.2) != 0.3. Produce a corrected decimal string with exactly 1 decimal place using toFixed.



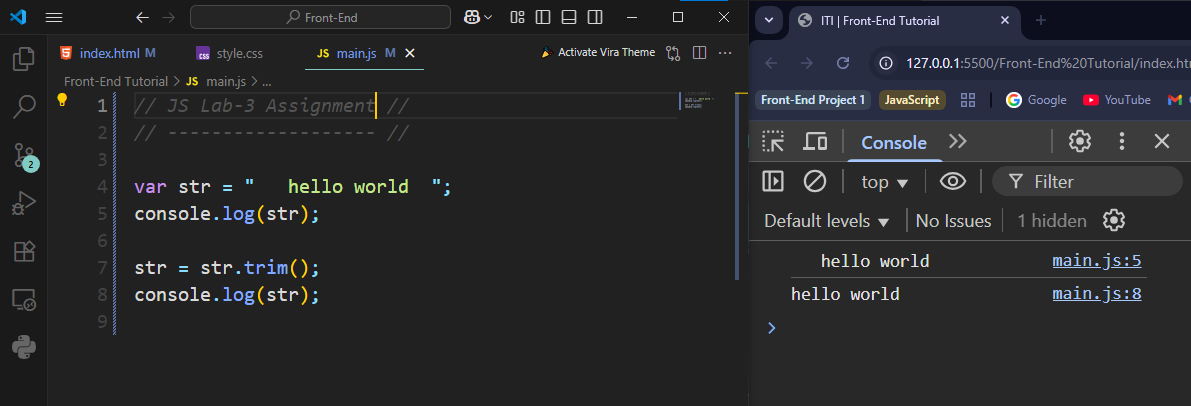
5. Write a function to safely parse a string that may contain trailing text (e.g. "120px") returning the integer part or null if it starts with non-digit.



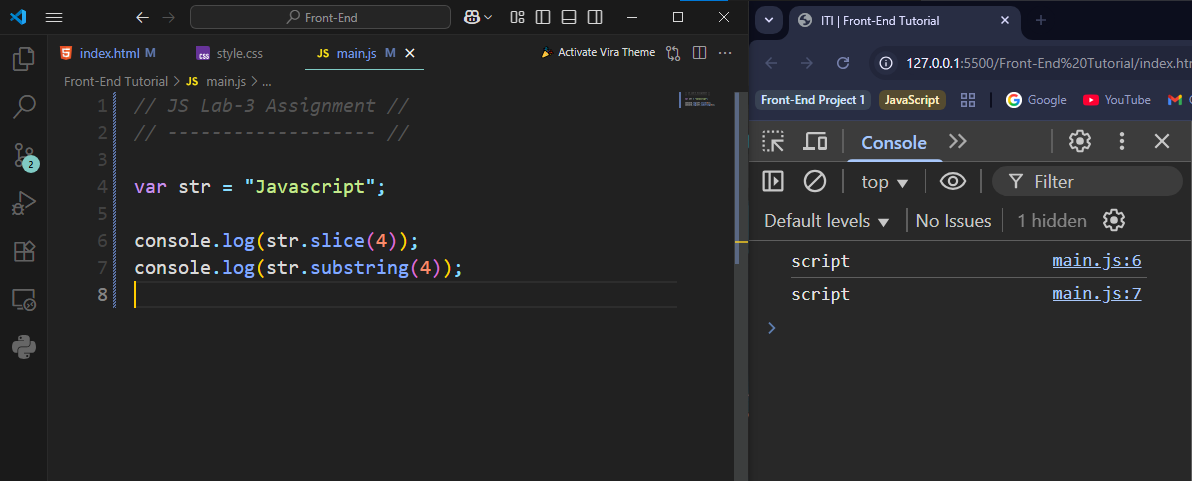
6. Implement isFiniteNumber(value) that returns true only for finite numeric values (reject numeric strings, Infinity, NaN, null, etc.) WITHOUT using Number.isFinite. Provide 4 passing and 4 failing test examples (comments).



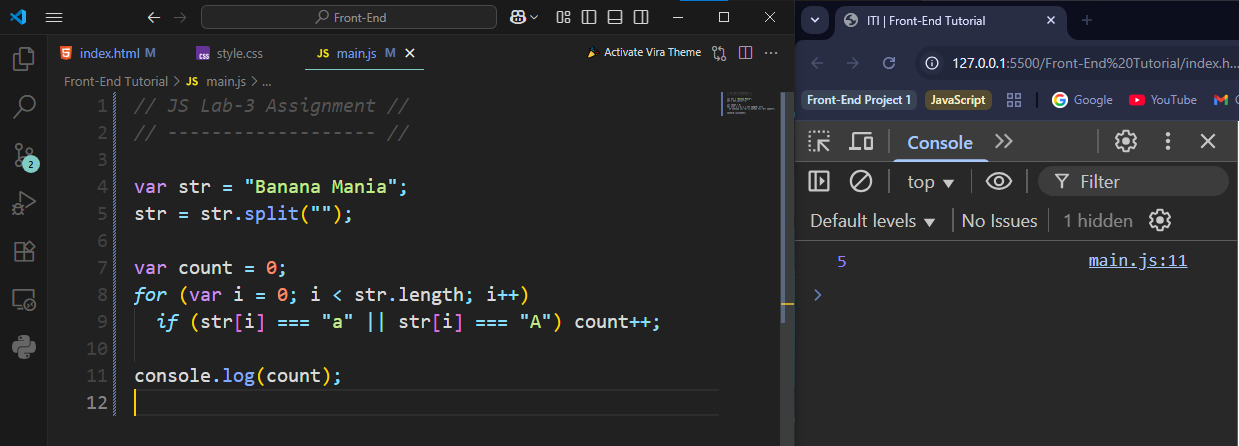
7. Remove leading and trailing spaces from the string " hello world ".



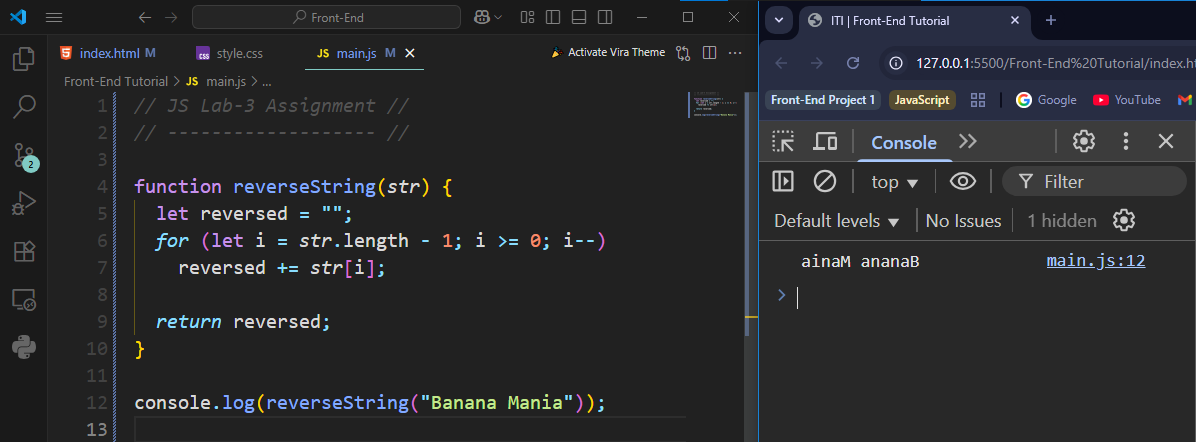
8. Get the substring "script" from "javascript" using two different methods (slice + substring).



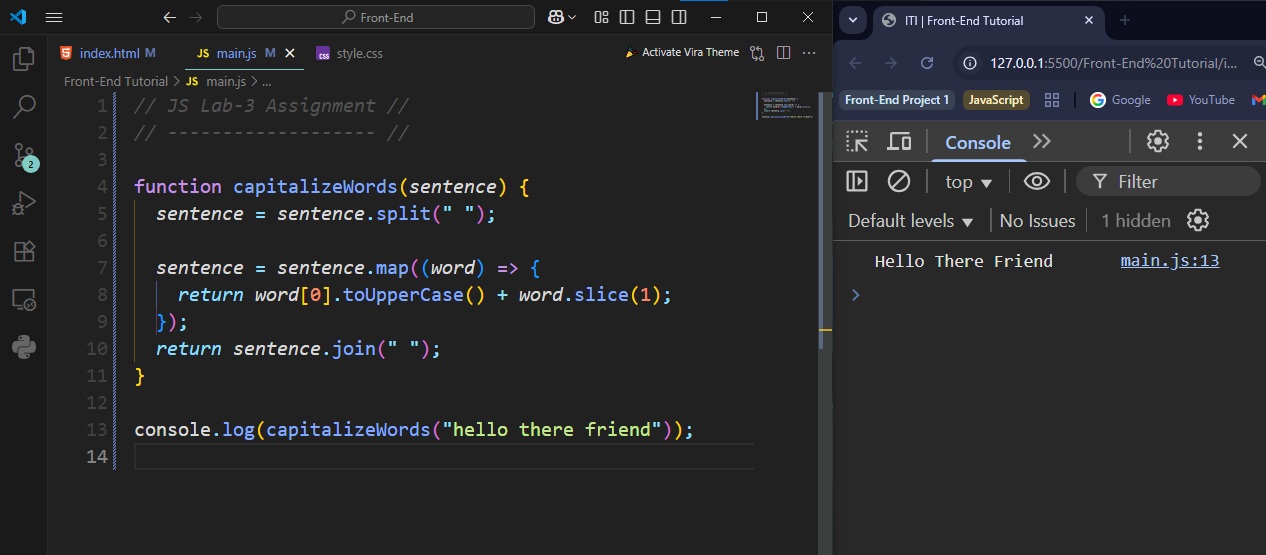
9. Count how many times the letter 'a' appears in "Banana Mania" (case-insensitive).



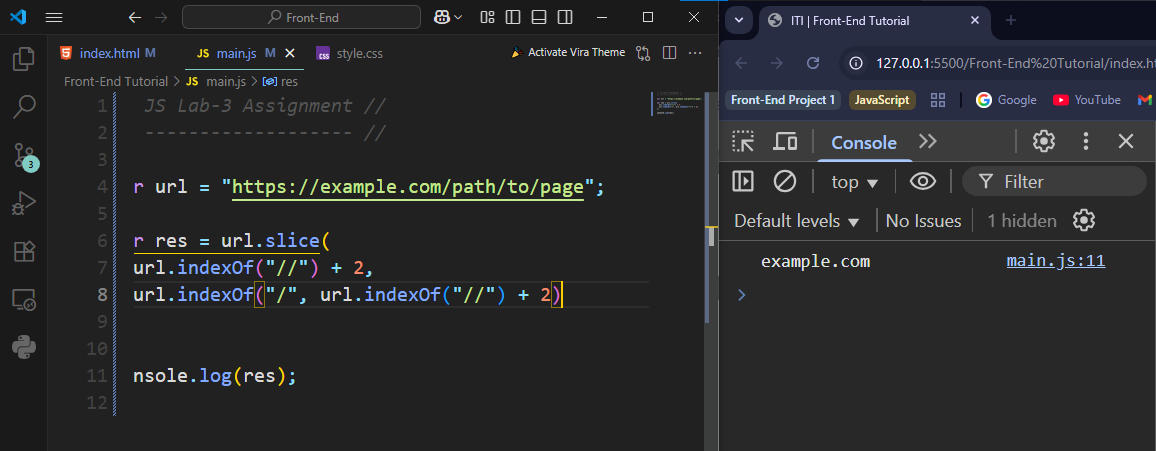
10. Write a function reverseString(s) without using array reverse (iterate backwards).



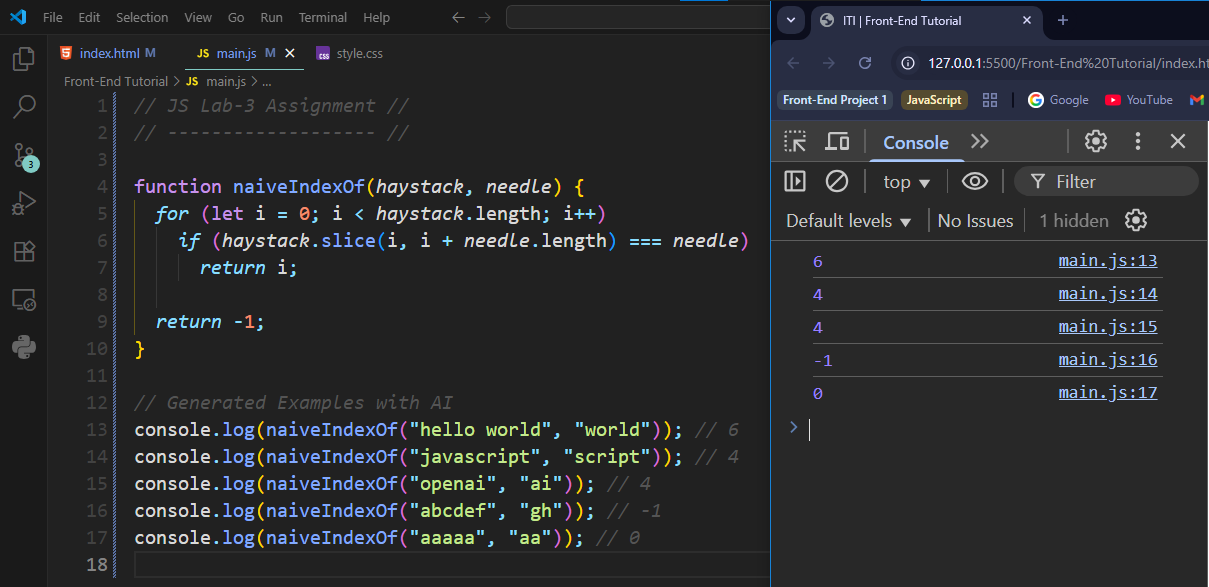
11. Write a function capitalizeWords(sentence) that turns "hello there friend" into "Hello There Friend".



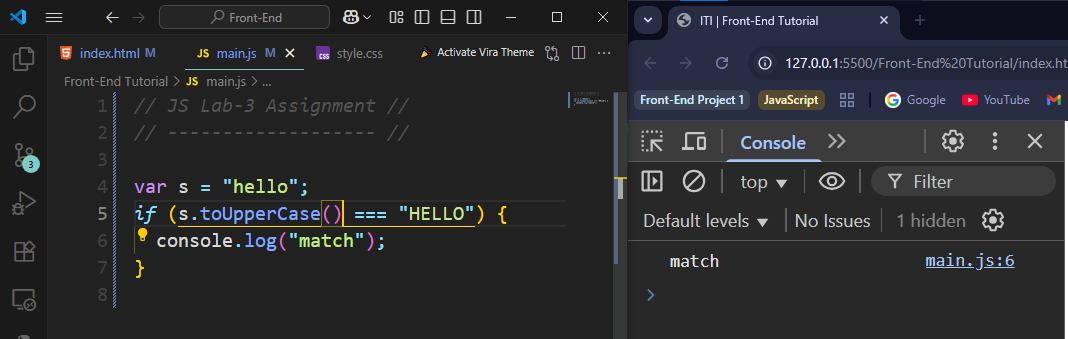
12. Extract the domain (without protocol and path) from a URL string like "https://example.com/path/to/page" (result: example.com) using indexOf + slice.



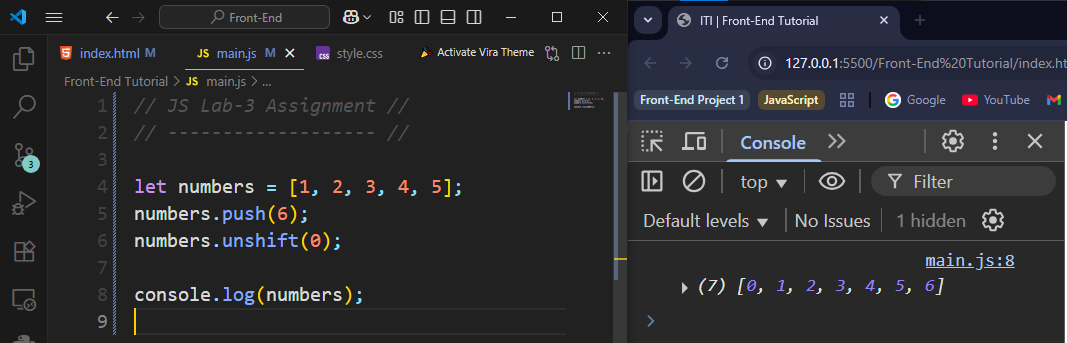
13. Implement a simple substring search function naiveIndexOf(haystack, needle) that returns first index or -1 (do not call built-in indexOf inside the loop).



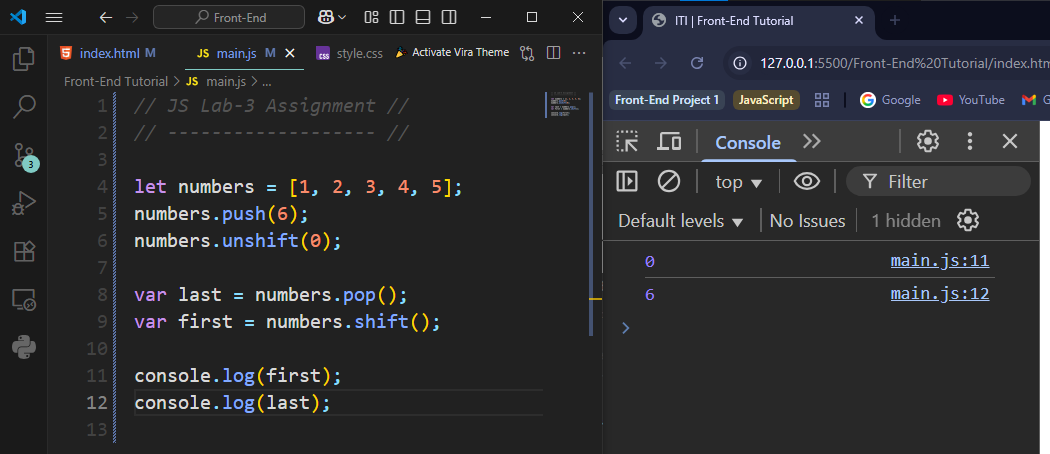
14. Buggy code: var s = 'hello'; if (s.toUpperCase = 'HELLO') { console.log('match'); } // Fix and explain issue.



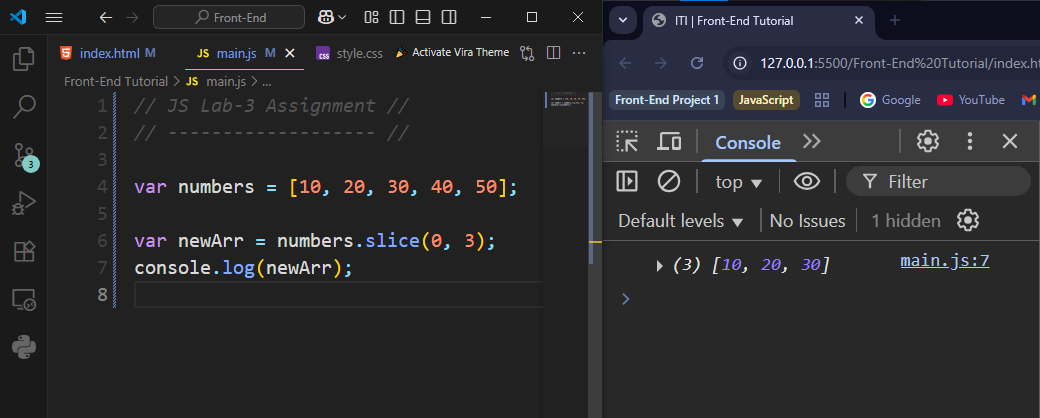
15. Create an array of the numbers 1..5, then push 6 and unshift 0.



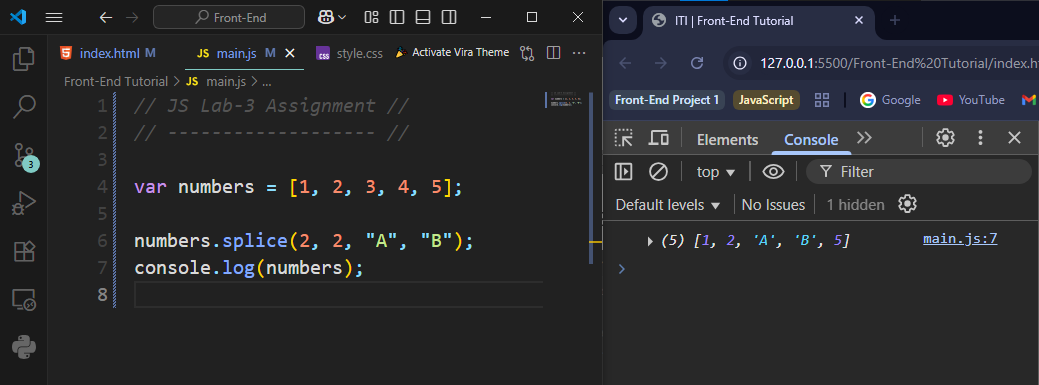
16. Remove the last element and store it. Remove the first element and store it.



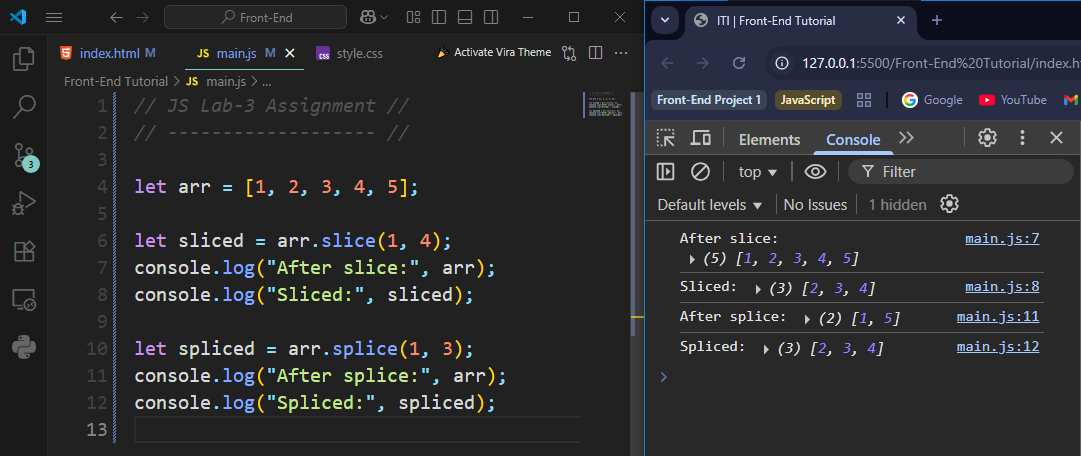
17. Use slice to copy the first 3 elements of [10,20,30,40,50] into a new array.



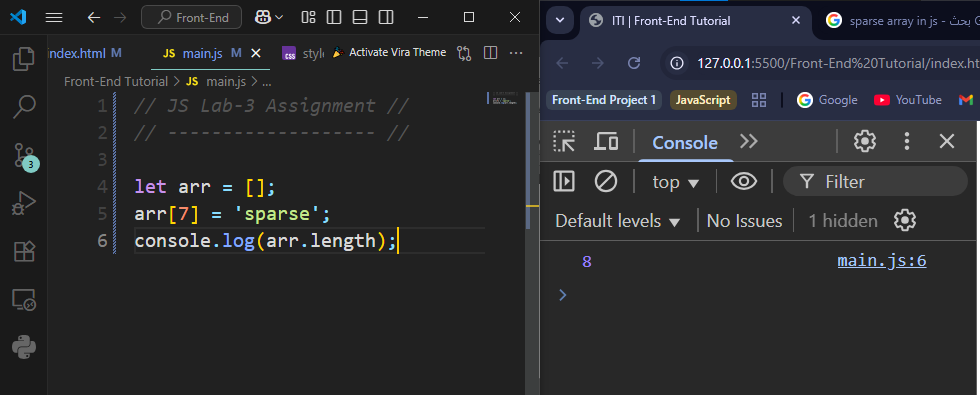
18. Use splice on [1,2,3,4,5] to remove 3 and 4 and insert 'a','b'. Result should be [1,2,'a','b',5].



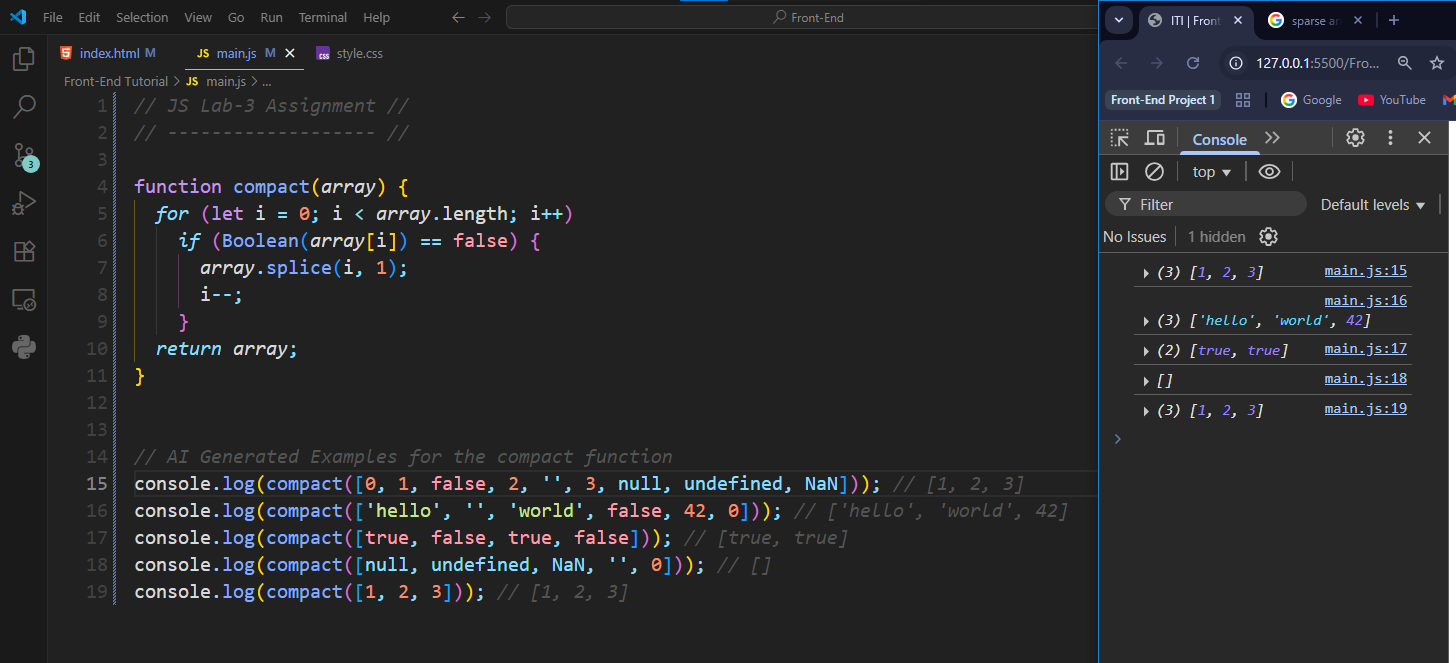
19. Demonstrate the difference between slice and splice on the same starting array (show original after each).



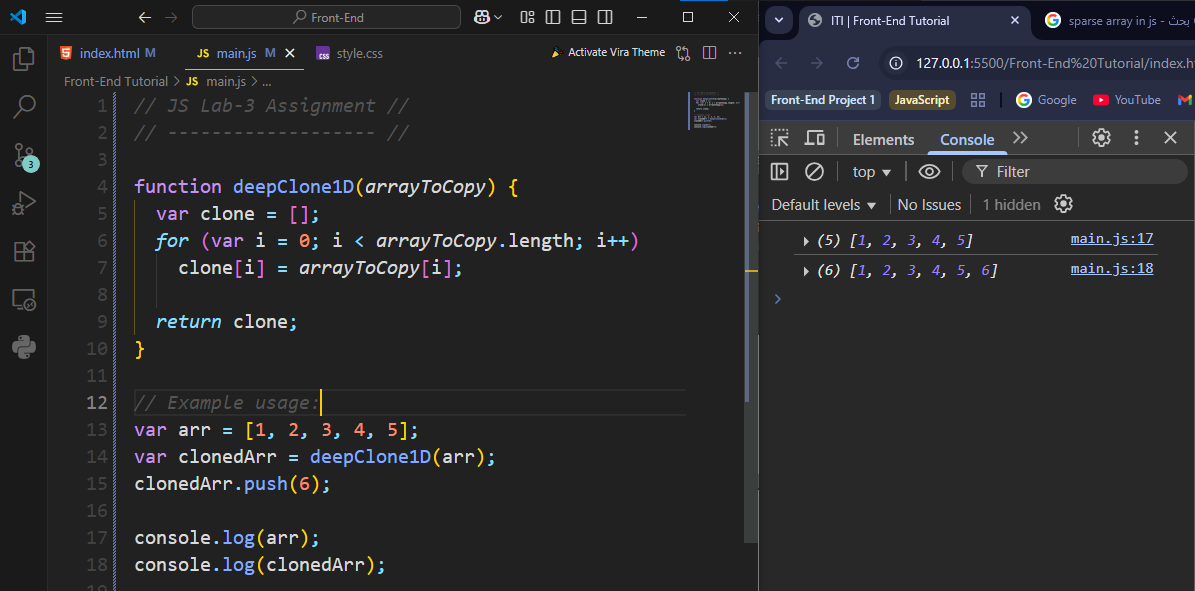
20. Create a sparse array by assigning index 7 on a fresh [] then log length.



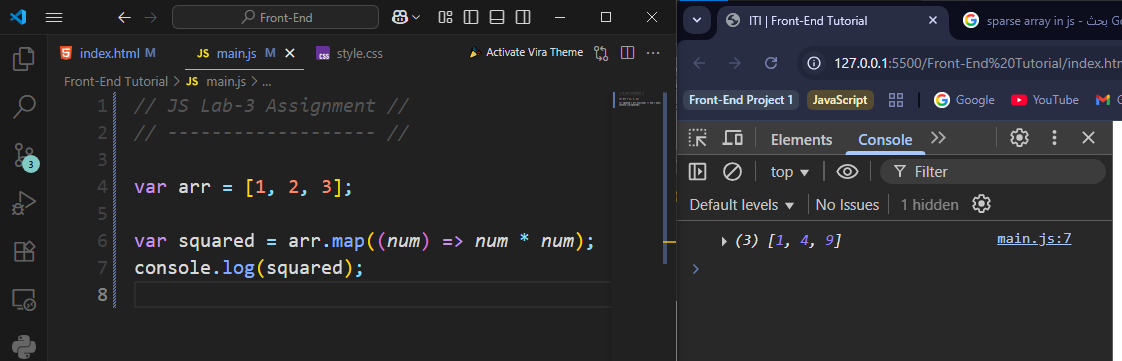
21. Write a function compact(array) that returns a new array without falsy values (manual loop, no filter).



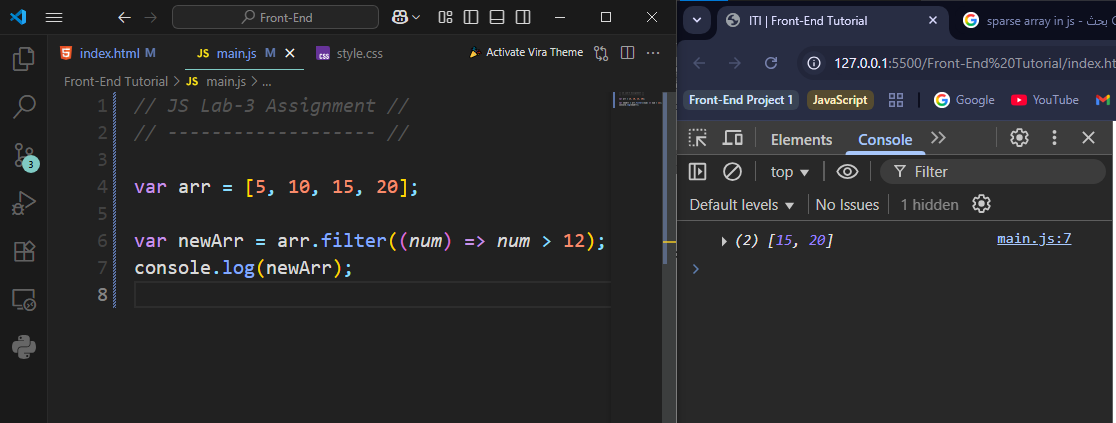
22. Implement a manual array clone deepClone1D(a) for a 1D array without using slice/concat .



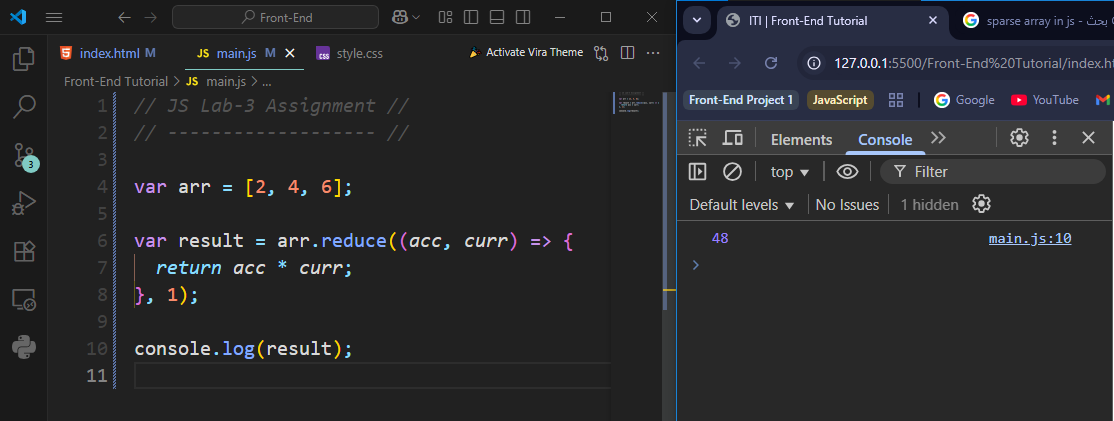
23. Map [1,2,3] to their squares using map.



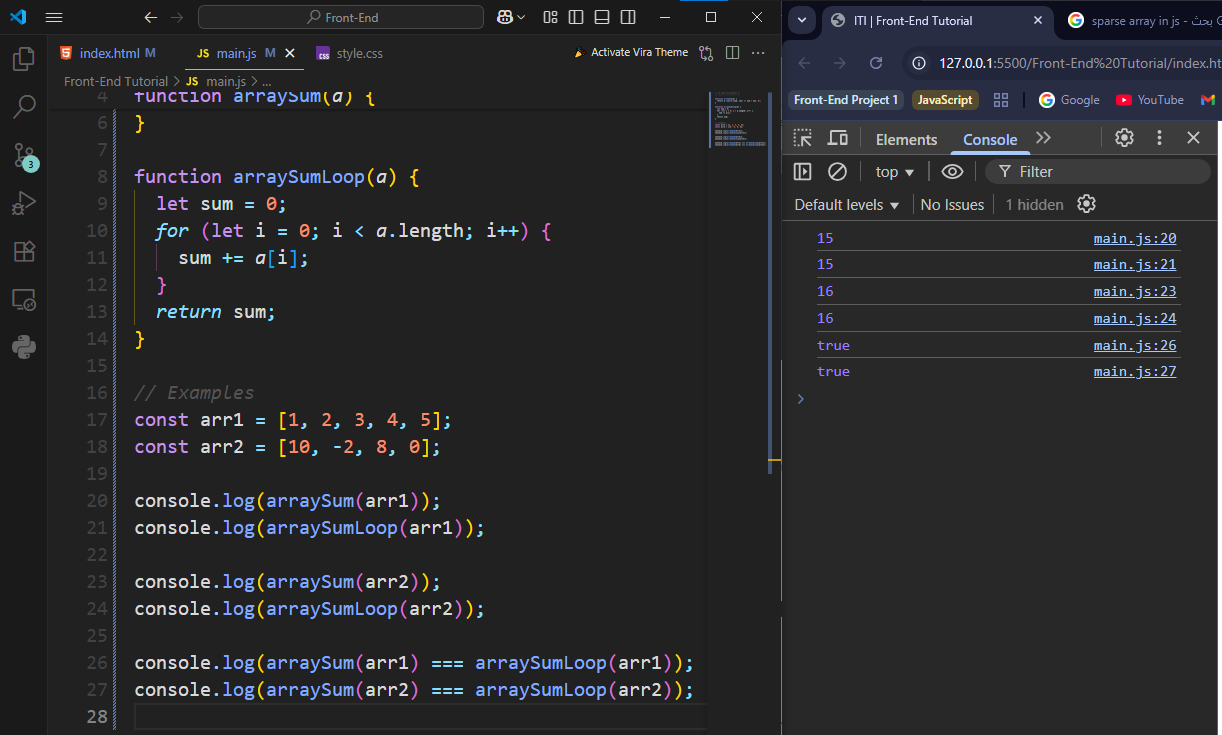
24. Filter [5,10,15,20] to keep values >= 12.



25. Reduce [2,4,6] to product.



26. Implement arraySum(a) using reduce; then implement arraySumLoop(a) using a for loop. Confirm outputs equal.



27. Given ['Ali','Sara','Kareem'] produce ['A','S','K'] (first letters) without using map (use for loop).

28. Implement unique(a) returning new array with duplicates removed (no ES6 Set). Complexity target: O(n^2) acceptable; comment how to improve.

29. Flatten one level: flatten1([1,[2,3],[4],5]) => [1,2,3,4,5] without using concat inside a loop (manual pushing and detection of Array).

31. Create object person with name and age; add a new property city after creation.

32. Access a property via bracket notation with a dynamic key variable.

33. Write function countKeys(obj) returning number of own enumerable properties (use for-in).

39. List (as comments) 5 different values that coerce to false in ES5.

40. safeToBoolean(v): return true only if v is strictly true, 'true', 1, or '1'; else false.

41. Create a Date for Jan 1, 2025 00:00 local.

42. Get the current year from new Date().

43. Write function daysBetween(d1, d2) returning whole day difference (ignore DST intricacies; ms/(1000\*60\*60\*24)).

44. Generate a random integer 1..100.

45. Round 4.567 to nearest integer, round down, and round up (three results).

46. randomIntArray(n, min, max): return array of length n with random ints (loop + push).

46. parsePriceList(str): Given "Apple:2.50;Orange:1.75;Banana:3" return object {Apple:2.5, Orange:1.75, Banana:3} (strings to numbers).

47. filterAndSortNumbers(mixedArray): keep only finite numbers then sort ascending (provide sample input and output). Use a numeric compare fn.