Problem Statement

Find the First Non-Repeating Character Write a program to find the first non-repeating character in a string. For input "swiss", the output should be "w". You cannot use any built-in string or character frequency counting functions.

type scripting program

```
function firstNonRepeatingCharacter(str: string): string {
 for (let i = 0; i < str.length; i++) {
  let isUnique = true;
  for (let j = 0; j < str.length; j++) {
   if (i !== j && str[i] === str[j]) {
     isUnique = false;
     break;
   }
  }
  if (isUnique) {
   return str[i];
  }
 }
 return ";
}
console.log(firstNonRepeatingCharacter("swiss")); // Output: "w"
```

Explanation

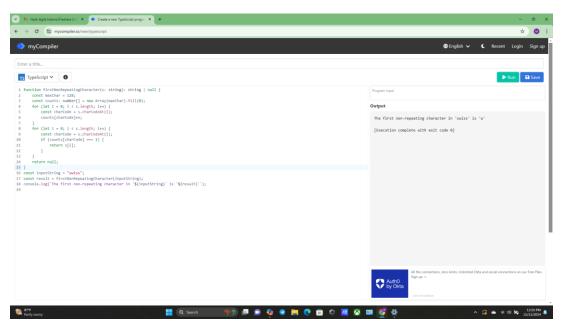
- **Function Definition:** firstNonRepeatingCharacter takes a string str to find the first non-repeating character.
- ② Outer Loop: Iterates through each character of str.
- Check Uniqueness: Initializes is Unique to true for each character.
- Inner Loop: Compares the current character with all others. If a match is found, isUnique becomes false, and the loop exits early.
- 2 Return Non-Repeating Character: If is Unique is still true, return the current character.
- 2 Return Empty String: If no non-repeating character is found, return an empty string.
- Time Complexity: O(n²) due to nested loops.
- 2 Space Complexity: O(1) because only a few variables are used.
- **Example:** For "swiss", the function returns "w" as it's the first non-repeating character.

Sample Inputs and Outputs

Sample 1:

Input: swiss

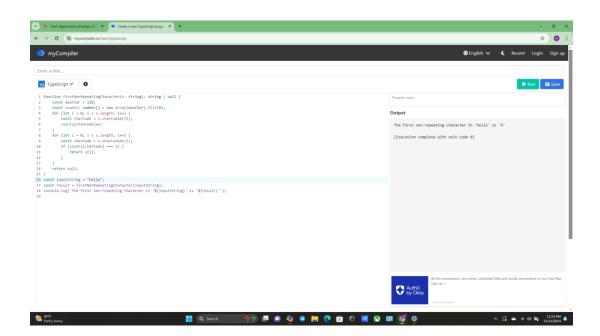
Output: w



Sample 2:

Input: hello

Output: h



Sample 3:

Input: concatenation

Output: e

