**Strings**

Strings are sequences of characters used to represent text.

Unlike character arrays, strings end with a special '\0' character in some languages like C.

In most languages like Java, Python, and JavaScript, strings are immutable, meaning they can’t be changed after creation. In C++, it is mutable.

You can access each character using its position (called an index), starting from 0.

So in "hello", the character at index 1 is 'e'.

# **Properties**

* Immutable (in many languages): Once created, their content cannot be changed (e.g., Python, JavaScript).
* Indexable: Each character can be accessed by its position (starting from 0).
* Iterable: You can loop through a string character by character.
* Support for Slicing/Substrings: You can extract parts of strings using ranges.
* Unicode Support: Most modern languages support Unicode, allowing international text.

# **String Operations**

## **Concatenation**

Concatenation is the addition of 2 strings and can be done in different languages in the following way:

* + Python:  
     str1 = "Hello"  
     str2 = "World"  
     result = str1 + " " + str2
  + C++:  
     string str1 = "Hello";  
     string str2 = "World";  
     string result = str1 + " " + str2;
  + Javascript:  
     let str1 = "Hello";  
     let str2 = "World";  
     let result = str1 + " " + str2;

## **Insertion**

* + Python:

s = "hello"

s = s[:2] + "X" + s[2:]

* + C++:

string s = "hello";

s.insert(2, "X");

* + Javascript:

let s = "hello";

s = s.slice(0, 2) + "X" + s.slice(2);// Inserts 'X' at s[2]

## **Deletion**

* + Python:

s = "hello"

s = s[:2] + s[3:] # Deletes character at index 2

* + C++

string s = "hello";

s.erase(2, 1); // Deletes 1 character at index 2

* + Javascript

let s = "hello";

s = s.slice(0, 2) + s.slice(3); // Deletes character at index 2

## **Length**

* + Python:  
     s = "hello"  
     print(len(s))
  + C++:  
     string s = "hello";  
     cout << s.length();
  + Javascript:  
     let s = "hello";  
     console.log(s.length);

## **Substring**

* + Python:  
     s = "hello"  
     print(s[1:4]) # "ell"
  + C++:  
     string s = "hello";  
     cout << s.substr(1,3); // "ell"
  + Javascript:  
     let s = "hello";  
     console.log(s.substring(1, 4)); // "ell"

## **Finding a Character**

* + Python:  
     s = "banana"  
     print(s.find("a"))
  + C++:  
     string s = "banana";  
     cout << s.find("a");
  + Javascript:  
     let s = "banana";  
     console.log(s.indexOf("a"));

## **Reversing a String**

* + Python:  
     s = "hello"  
     print(s[::-1])
  + C++:  
     #include <algorithm>  
     string s = "hello";  
     reverse(s.begin(), s.end());
  + Javascript:  
     let s = "hello";  
     console.log(s.split("").reverse().join(""));

You can also visit:

[String in Data Structure | GeeksforGeeks](https://www.geeksforgeeks.org/string-data-structure/)

[JavaScript Strings](https://www.w3schools.am/js/js_strings.html#gsc.tab=0)

[Python Strings](https://www.w3schools.in/python/strings/)

[C++ Strings](https://www.w3schools.com/cpp/cpp_strings.asp)

# **Problems**

Easy:

[Find the Index of the First Occurrence in a String - LeetCode](https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string/description/) | [Solution](https://algo.monster/liteproblems/28)

[Valid Anagram - LeetCode](https://leetcode.com/problems/valid-anagram/description/) | [Solution](https://www.geeksforgeeks.org/check-whether-two-strings-are-anagram-of-each-other/)

[Roman to Integer - LeetCode](https://leetcode.com/problems/roman-to-integer/description/?envType=problem-list-v2&envId=string) | [Solution](https://www.geeksforgeeks.org/roman-number-to-integer/)

[Valid Palindrome - LeetCode](https://leetcode.com/problems/valid-palindrome/description/?envType=problem-list-v2&envId=string) | [Solution](https://www.geeksforgeeks.org/sentence-palindrome-palindrome-removing-spaces-dots-etc/)

[Isomorphic Strings - LeetCode](https://leetcode.com/problems/isomorphic-strings/description/?envType=problem-list-v2&envId=string) | [Solution](https://www.geeksforgeeks.org/check-if-two-given-strings-are-isomorphic-to-each-other/)

[Number of Segments in a String - LeetCode](https://leetcode.com/problems/number-of-segments-in-a-string/description/?envType=problem-list-v2&envId=string) | [Solution](https://algo.monster/liteproblems/434)

[Repeated Substring Pattern - LeetCode](https://leetcode.com/problems/repeated-substring-pattern/description/?envType=problem-list-v2&envId=string) | [Solution](https://algo.monster/liteproblems/459)

Medium:

[Fraction to Recurring Decimal - LeetCode](https://leetcode.com/problems/fraction-to-recurring-decimal/description/?envType=problem-list-v2&envId=string) | [Solution](https://www.geeksforgeeks.org/represent-the-fraction-of-two-numbers-in-the-string-format/)

[Longest Palindromic Substring - LeetCode](https://leetcode.com/problems/longest-palindromic-substring/description/) | [Solution](https://www.geeksforgeeks.org/longest-palindromic-substring/#using-expansion-from-center)

[Find All Anagrams in a String - LeetCode](https://leetcode.com/problems/find-all-anagrams-in-a-string/description/?envType=problem-list-v2&envId=string) | [Solution](https://algo.monster/liteproblems/438)

[Longest Substring Without Repeating Characters - LeetCode](https://leetcode.com/problems/longest-substring-without-repeating-characters/description/) | [Solution](https://www.geeksforgeeks.org/length-of-the-longest-substring-without-repeating-characters/)

[Longest Repeating Character Replacement - LeetCode](https://leetcode.com/problems/longest-repeating-character-replacement/description/?envType=problem-list-v2&envId=string) | [Solution](https://algo.monster/liteproblems/424)

Hard:

[Substring with Concatenation of All Words - LeetCode](https://leetcode.com/problems/substring-with-concatenation-of-all-words/description/?envType=problem-list-v2&envId=string) | [Solution](https://algo.monster/liteproblems/30)

[Smallest K-Length Subsequence With Occurrences of a Letter - LeetCode](https://leetcode.com/problems/smallest-k-length-subsequence-with-occurrences-of-a-letter/description/) | [Solution](http://geeksforgeeks.org/lexicographically-smallest-k-length-subsequence-from-a-given-string/)

[Minimum Window Substring - LeetCode](https://leetcode.com/problems/minimum-window-substring/description/?envType=problem-list-v2&envId=string) | [Solution](https://algo.monster/liteproblems/76)