

# Strings – String Traversal

## 1. What is String Traversal?

**String traversal** is the process of **accessing each character of a string one by one**, usually from the **first character to the last character**, in order to perform some operation on it.

In simple terms, it means **visiting every character in the string sequentially**.

---

## 2. Why is String Traversal Important?

String traversal is important because:

- Almost all string operations depend on traversal
- It is required for comparison, searching, and modification tasks
- It helps in analyzing and processing textual data
- It forms the base for advanced string algorithms

Without traversal:

- We cannot count characters
  - We cannot reverse a string
  - We cannot check palindromes or frequencies
- 

## 3. How Does String Traversal Work?

String traversal is typically performed using a **loop** that starts from the **first index (0)** and moves till the **last index (length – 1)**.

### General Logic:

1. Start from index 0
2. Access the character at the current index
3. Perform the required operation

4. Move to the next index
  5. Stop when the last character is reached
- 

## 4. Visual Representation of String Traversal

Consider the string:

"CODE"

Index	0	1	2	3
Char	C	O	D	E

Traversal flow:

Start → 'C' → 'O' → 'D' → 'E' → Stop

A pointer moves **left to right**, visiting each character exactly once.

---

## 5. Types of String Traversal

### 1 Forward Traversal

- Starts from the first character
- Ends at the last character
- Most commonly used

### 2 Reverse Traversal

- Starts from the last character
  - Ends at the first character
  - Used in string reversal and backward checks
- 

## 6. Common Operations Using String Traversal

String traversal is used to:

- Print characters of a string
  - Count vowels and consonants
  - Check palindrome
  - Reverse a string
  - Count frequency of characters
  - Search for a specific character
- 

## 7. Time and Space Complexity

- **Time Complexity:**  $O(n)$   
(Each character is visited once)
  - **Space Complexity:**  $O(1)$   
(Only a loop variable is used)
- 

## 8. Advantages of String Traversal

- Simple and easy to implement
  - Works for all string sizes
  - Essential for text processing
  - Efficient for sequential operations
- 

## 9. Limitations of String Traversal

- Sequential access only
  - Cannot skip characters randomly
  - Inefficient if repeated unnecessarily
- 

## 10. Real-World Examples

- Counting letters in a word
  - Checking password strength
  - Validating email formats
  - Searching keywords in text
  - Text analytics and parsing
- 

## 11. Summary

- String traversal means visiting each character
  - Uses index-based access
  - Loop-based approach is used
  - Time complexity is  $O(n)$
  - Forms the foundation for all string operations
-