

Session 2

Strings

1000@gmail.com



CONTENTS

1. Array of Characters
2. Similarities with Arrays
3. Common String Methods
4. Advantages of String Operations
5. Common Terminologies in Strings
6. Common String Operations
7. Activity



01

Array of Characters

pandiyanalex1000@gmail.com



• Introduction to Strings

Definition

An array of characters is essentially a string, which is a sequence of characters treated as a single data structure.

Properties

Alphanumeric and Special Characters

Can contain alphanumeric characters (e.g., A, 1, Z) and special characters (e.g., @, #, !).

Immutability

Strings are immutable in many programming languages, meaning they cannot be changed once created (though operations can create new strings).



02

Similarities with Arrays



Indexing

Accessing Characters

Like arrays, strings support indexing, allowing access to specific characters.

Example:

String : H E L L O
0 1 2 3 4

Length

Strings, like arrays, have a length property.

Example:

"Hello" length gives 5.

Iterability

Looping Through

You can loop through a string character by character, similar to iterating through an array.



03

Common String Methods

dyana100@gmail.com



• Methods Similar to Arrays

includes(value)

Checks if the string contains a specific value.

Example "Hello".includes("ll") → true

indexOf(value)

Returns the index of the first occurrence of a value.

Example "Hello".indexOf("e") → 1

slice(start, end)

Extracts a portion of the string (non-mutative).

Example "Hello".slice(1, 4) → "ell"



split(separator)

Splits the string into an array based on the specified separator.

Example "Hello World".split(" ") → ["Hello", "World"]

join(separator)

Can reassemble split strings by joining array elements into a single string.

Example ["Hello", "World"].join(" ") → "Hello World"



• Methods in both Java and Python

Functionality	Java Method	Python Method
Length of String	<code>str.length()</code>	<code>len(str)</code>
Character at Index	<code>str.charAt(index)</code>	<code>str[index]</code>
Substring	<code>str.substring(start, end)</code>	<code>str[start:end]</code>
Lowercase	<code>str.toLowerCase()</code>	<code>str.lower()</code>
Uppercase	<code>str.toUpperCase()</code>	<code>str.upper()</code>
Trim Spaces	<code>str.trim()</code>	<code>str.strip()</code>
Replace Characters	<code>str.replace("a", "b")</code>	<code>str.replace("a", "b")</code>
Split String	<code>str.split(" ")</code>	<code>str.split(" ")</code>
Join String	<code>String.join(", ", arr)</code>	<code>" ".join(arr)</code>
Check Prefix	<code>str.startsWith("abc")</code>	<code>str.startswith("abc")</code>
Check Suffix	<code>str.endsWith("xyz")</code>	<code>str.endswith("xyz")</code>
Find Index	<code>str.indexOf("x")</code>	<code>str.find("x")</code>
Contains Substring	<code>str.contains("xyz")</code>	<code>"xyz" in str</code>



04

Advantages of String Operations





• Useful Applications

Enable powerful manipulation of text data.

Useful for:

1. Parsing files.
2. Formatting user input.
3. Handling large text-based data efficiently.



05

Common Terminologies in Strings

Pandiyanalex10



Substring vs Subsequence

Substring

A substring is a continuous part of a string.
Extracting a substring involves taking out characters in a continuous order between two indices.

Key Features:

1. Must maintain contiguity (no breaks).
2. A substring is always a part of the original string.

Example:

"quick" in the string "thequickbrownfox".

Subsequence

Formed by removing zero or more characters without changing the order of the remaining characters.

Key Features:

1. Does not require contiguity.
2. Any sequence of characters that preserves the original order can be a subsequence.

Example:

"qck" in the string "thequickbrownfox".



Prefix vs Suffix

Prefix

A prefix is a substring that occurs at the beginning of the string.

Key Features:

1. Starts from the first character.
2. Can be of any length (including an empty string).

Example:

Given string: "programming"

"Prefix: "prog" (substring starting from index 0 - 3).

Suffix

A suffix is a substring that occurs at the end of the string.

Key Features:

1. Ends with the last character.
2. Can also be of any length.

Example:

Given string: "programming"

Suffix: "ming" (substring ending at the last character).



• Mutable vs Immutable

Mutable

An object is mutable if its value can be changed after it has been initialized.

Key Features:

Changes directly impact the original object.

Immutable

An object is immutable if its value cannot be changed after it has been initialized.

Key Features:

Any modification creates a new object, leaving the original unchanged.



06

Common String Operations



Common String Operations

1. Reversing a String:

Show "hello" turning into "olleh".

2. Check if Identical:

Compare two strings side-by-side with a case-sensitive comparison.

3. Sorting:

Show "cabfed" being rearranged to "abcdef".

4. Concatenation:

Combine "Hello" and "World" visually with a + symbol.

5. Count Characters:

Display total length and count of specific characters in a string.

6. Removing Characters:

Show unwanted characters being "crossed out" (e.g., "hello" → "heo").

7. Substrings and Subsequences:

Use diagrams to demonstrate continuous (substring) vs non-continuous (subsequence) extraction.



07

Activity



• Activity

- </> Capitalize the First letter of each word - In Session
- </> Valid Palindrome
- </> Reverse words in a String





Activity

</> Reverse words in a String - In Session

</> String Compress

</> Reverse only Letters

