Table of contents

Strings

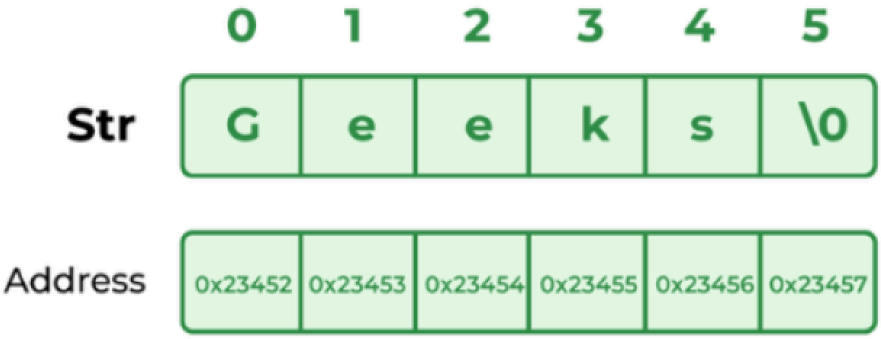
Strings are sequences of characters. The differences between a character array and a string are that a string is terminated with a special character '\0' and strings are typically immutable in most programming languages like Java, Python and JavaScript. Examples are:

"Hello World", "Pakistan", "City", etc.

How are Strings represented in Memory?

In C, a string can be referred to either using a character pointer or as a character array. When strings are declared as character arrays, they are stored like other types of arrays in C. String literals (assigned to pointers) are immutable in C and C++.

In C++, strings created using the string class are mutable and internally represented as arrays. In Python, Java, and JavaScript, string characters are stored at contiguous locations (like arrays).



	0	1	2	3	4	5
Str	G	e	e	k	s	\0
Address	0x23452	0x23453	0x23454	0x23455	0x23456	0x23457

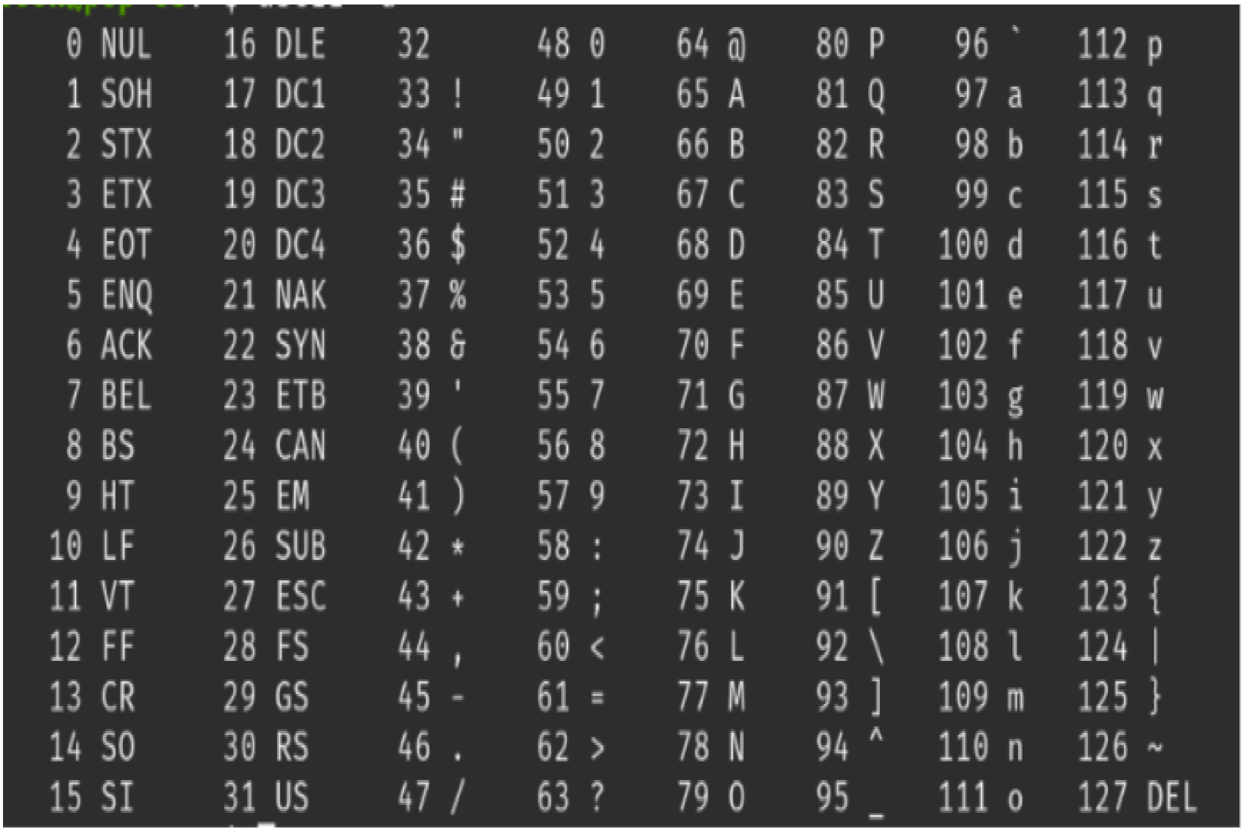
General Operations performed on String

Here we are providing you with some must-know concepts of string:

- **Length of String:** The length of a string refers to the total number of characters present in it, including letters, digits, spaces, and special characters. It is a fundamental property of strings in any programming language and is often used in various operations such as validation, manipulation, and comparison.

- **Search a Character:** Searching for a character in a string means finding the position where a specific character appears. If the character is present multiple times, you might need to find its first occurrence, last occurrence, or all occurrences.
- **Check for Substring:** Checking for a substring means determining whether a smaller sequence of characters exists within a larger string. A substring is a continuous part of a string, and checking for its presence is a common operation in text processing, search algorithms, and data validation.
- **Insert a Character**: Inserting a character into a string means adding a new character at a specific position while maintaining the original order of other characters. Since strings are immutable in many programming languages, inserting a character usually involves creating a new modified string with the desired character placed at the specified position.
- **Delete a Character**: Deleting a character from a string means removing a specific character at a given position while keeping the remaining characters intact. Since strings are immutable in many programming languages, this operation usually involves creating a new string without the specified character.
- **Check for Same Strings**: Checking if two strings are the same means comparing them character by character to determine if they are identical in terms of length, order, and content. If every character in one string matches exactly with the corresponding character in another string, they are considered the same.
- **String Concatenation**: String concatenation is the process of joining two or more strings together to form a single string. This is useful in text processing, formatting messages, constructing file paths, or dynamically creating content.
- **Reverse a String:** Reversing a string means arranging its characters in the opposite order while keeping their original positions intact in the reversed sequence. This operation is commonly used in text manipulation, data encryption, and algorithm challenges.
- **Rotate a String:** Rotating a string means shifting its characters to the left or right by a specified number of positions while maintaining the order of the remaining characters. The characters that move past the boundary wrap around to the other side.
- **Check for Palindrome**: Checking for a palindrome means determining whether a string reads the same forward and backward. A palindrome remains unchanged when reversed, making it a useful concept in text processing, algorithms, and number theory.

ASCII Table



```
  0 NUL   16 DLE   32      48 0   64 @   80 P    96 `   112 p
  1 SOH   17 DC1   33 !    49 1   65 A   81 Q    97 a   113 q
  2 STX   18 DC2   34 "    50 2   66 B   82 R    98 b   114 r
  3 ETX   19 DC3   35 #    51 3   67 C   83 S    99 c   115 s
  4 EOT   20 DC4   36 $    52 4   68 D   84 T   100 d   116 t
  5 ENQ   21 NAK   37 %    53 5   69 E   85 U   101 e   117 u
  6 ACK   22 SYN   38 &    54 6   70 F   86 V   102 f   118 v
  7 BEL   23 ETB   39 '    55 7   71 G   87 W   103 g   119 w
  8 BS    24 CAN   40 (    56 8   72 H   88 X   104 h   120 x
  9 HT    25 EM    41 )    57 9   73 I   89 Y   105 i   121 y
 10 LF    26 SUB   42 *    58 :   74 J   90 Z   106 j   122 z
 11 VT    27 ESC   43 +    59 ;   75 K   91 [   107 k   123 {
 12 FF    28 FS    44 ,    60 <   76 L   92 \   108 l   124 |
 13 CR    29 GS    45 -    61 =   77 M   93 ]   109 m   125 }
 14 SO    30 RS    46 .    62 >   78 N   94 ^   110 n   126 ~
 15 SI    31 US    47 /    63 ?   79 O   95 _   111 o   127 DEL
```

C++ Built-In String Functions

Function	Description
size()	Give the size of the string.
resize()	Resizes the string.
find()	Finds the substring and returns the position.
rfind()	Finds the last occurrence of the substring.
substr()	Returns a substring.
compare()	Compares the two strings.
replace()	Replaces part of a string.
erase()	Erases part of a string.
clear()	Removes all the characters.