

String

- String is **inbuilt class** in C++ and also can be used as a **data type** to store text between double quotes

→ `String name = "Ahmed";`

→ In this line string behaves as a **data type**

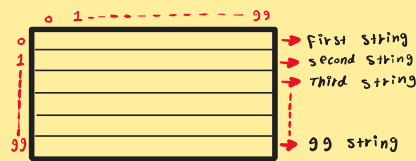
- String is actually behaves like a **character array** but in character array we have to take care of **NULL character '\0'** which not found in the string.

- How you can store a **MULTIPLE** string?

1] using 2D Array

→ `Char name[100][100];`

→ Note: This way is not the better way



2] using Vectors

→ `Vector<string> vstring;`

→ `V.push_back("Ahmed");`

→ `V.push_back("Ali ");`

→ `V.push_back("Mona");`



```
for (int i = 0; i < vstring.size(); i++)  
{  
    cout << vstring[i] << " ";  
    sort(vstring[i].begin(), vstring[i].end());  
    cout << vstring[i] << endl;  
}
```

How to use `getline` with string

→ `String str;`

→ `getline(cin, str);` // "Ahmed"

→ now you can use spaces in your input

After taking the input you can treat with the string as an array

→ `cout << str[0] << endl;` // A

→ `cout << str[1] << endl;` // H

→ `cout << str[2] << endl;` // M

1. `length()` / `size()`: Returns the length of the string.
2. `empty()`: Checks if the string is empty.
3. `clear()`: Clears the contents of the string.
4. `append()` / `+=`: Appends a string or a character to the end of the string.
5. `insert()`: Inserts characters into the string at a specified position.
6. `erase()`: Erases characters from the string.
7. `replace()`: Replaces a portion of the string with another string.
8. `substr()`: Extracts a substring from the string.
9. `find()` / `rfind()`: Searches for a substring within the string.
10. `compare()`: Compares two strings.
11. `at()` / `[]`: Accesses individual characters in the string.
12. `c_str()`: Returns a pointer to a null-terminated character array representing the string.
13. `resize()`: Resizes the string.
14. `swap()`: Swaps the contents of two strings.
15. `getline()`: Reads a line from an input stream into the string.