C++ String

A `string` variable contains a collection of characters surrounded by double quotes (""):

```
#include<string>
string greeting = "Hello";
```

Note:

Some compilers (especially GCC or Clang) **automatically include <string> internally** when you use <iostream>.
That's because <iostream> depends on <ios> → <istream> → <ostream> → and *these headers internally include parts of <string>*.

So, by accident (or convenience), std::string becomes *visible* to your program — **even if you didn't include <string> explicitly**.

String Concatenation

```
string firstName = "Ghada";
string lastName = "Mohamed";
string fullName = firstName + lastName;
cout << fullName;
```

output:

```
GhadaMohamed
```

```
string firstName = "John";
string lastName = "Doe";
string fullName = firstName + " " + lastName;
cout << fullName;
```

output:

```
Ghada Mohamed
```

Adding Numbers and Strings

Note:

C++ uses the + operator for both **addition** and **concatenation**.

Numbers are added. Strings are concatenated.

Int Example:

```
int x = 10;
int y = 20;
int z = x + y;      // z will be 30 (an integer)
```

String Example:

```
string x = "10";
string y = "20";
string z = x + y;   // z will be 1020 (a string)
```

If you try to add a number to a string, an error occurs:

```
string x = "10";
int y = 20;
string z = x + y;
```

String Length

To get the length of a string, use the `length()` or `size()` function:

```
string txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
cout << "The length of the txt string is: " << txt.length();
```

output:

```
The length of the txt string is: 26
```

Strings - Special Characters

Because strings must be written within quotes, C++ will misunderstand this string, and generate an error:

```
string txt = "We are the so-called "Vikings" from the north.";
```

The solution to avoid this problem, is to use the **backslash escape character**.

The backslash (\) escape character turns special characters into string characters:

Escape character	Result	Description
\'	'	Single quote
\"	"	Double quote
\\	\	Backslash

Example:

```
string txt = "We are the so-called \"Vikings\" from the north.";
string txt = "It\'s alright.";
```

Access Strings

you can access the characters in a string by referring to its index number inside square brackets `[]`.

```
string myString = "Hello";
cout << myString[0];
// Outputs H
```

```
cout << myString[myString.length() - 1];
// Outputs o
```

Change String Characters

To change the value of a specific character in a string, refer to the index number, and use single quotes:

```
string myString = "Hello";
myString[0] = 'J';
cout << myString;
// Outputs Jello instead of Hello
```