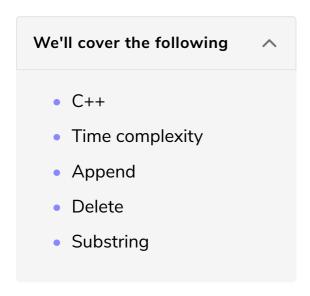# STL

In this lesson, we'll see how to use string in C++.

C++

Though strings can be represented as an array of characters, most languages have a separate type for strings.

Include statement: `#include <string>`

Let's see the string type in C++ and some of its methods that we will be using frequently.

The entire documentation can be found here.

```
string s; // new empty string

s.size(); // get size()
s[i]; // access character
```

Time complexity

The string data type is an array of characters and hence the operations are very similar to that on a vector; memory, time and name-wise. Points to note are

- Inserting at the end - $O(1)$
- Inserting in between - $O(N)$

- Deleting last character - $O(1)$

- Deleting other characters - $O(N)$

A quick recap, inserting or deleting middle characters requires the shifting of other characters.

Append

`string` has multiple methods to append to an existing string.

- $append(s)$ - append another string
- `+=` $operator$ - append another string
- $push\_back(c)$ - append character

```
string s1;
string s2 = "str";

s1.append(s2); // s1 = "str"
s1.push_back('i'); // s1 = "stir"
s1+="ng"; // s1 = "string"
```

Delete

- $erase(it)$ - Iterator to the character to remove
- $erase(pos, len)$ - erase `len` number of characters starting at position `pos`

```
string s = "example string";

s.erase(s.begin() + 7); // s = "examplestring"
s.erase(4, 6); // s = "examing"
```

Substring

The substring `substr(start_pos, len)` returns substring for passed start position and length (length is optional, if skipped, return the string till the end).

Here is a code snippet where the functions return `lo W`. Fix it to return `Hell` to pass the test.

```cpp
#include <iostream>
#include <string>

using namespace std;

string substring(string s) {
  return s.substr(3, 4);
}
```

That's all we need for easy to easy-medium string problems. We'll discuss string algorithms in later lessons. Let's discuss a solved problem in the next lesson.

```cpp
#include <iostream>
#include <string>

using namespace std;

string substring(string s) {
  return s.substr(3, 4);
}
```