

SALE!

✕

GeeksforGeeks Courses Upto 25% Off Enroll Now!

[Save 25% on Courses](#) [DSA](#) [Data Structures](#) [Algorithms](#) [Interview Preparation](#) [Data Science](#) [T](#)

Raw String Literal in C++

Difficulty Level : Easy • Last Updated : 27 Jan, 2023

[Read](#) [Discuss](#) [Courses](#) [Practice](#) [Video](#)

A Literal is a constant variable whose value does not change during the lifetime of the program. Whereas, a raw string literal is a string in which the escape characters like '`\n`', '`\t`', or '`\"`' of C++ are not processed. Hence, a raw string literal that starts with **R** (and ends in **)**.

The syntax for Raw string Literal:

```
R "delimiter( raw_characters )delimiter" // delimiter is the end of logical entity
```

Here, delimiter is optional and it can be a character except the backslash{ / }, whitespaces{ }, and parentheses { () }.

These raw string literals allow a series of characters by writing precisely its contents like raw character sequence.

AD

Example:

Ordinary String Literal

```
"\\n"
```

Raw String Literal

```
    \/-- Delimiter  
R"(\n)"  
    /\-- Delimiter
```

Difference between an Ordinary String Literal and a Raw String Literal:

Ordinary String Literal	Raw String Literal
It does not need anything to be defined.	It needs a defined line{ parentheses ()} to start with the prefix R .
It does not allow/include nested characters.	It allows/includes nested character implementation.
It does not ignore any special meaning of character and implements their special characteristic.	It ignores all the special characters like \n and \t and treats them like normal text.

Example of Raw String Literal:

CPP

```
// C++ program to demonstrate working of raw string literal  
#include <iostream>  
using namespace std;  
  
// Driver Code  
int main()  
{  
    // A Normal string  
    string string1 = "Geeks.\nFor.\nGeeks.\n";  
  
    // A Raw string  
    string string2 = R"(Geeks.\nFor.\nGeeks.\n)";  
  
    cout << string1 << endl;  
  
    cout << string2 << endl;  
  
    return 0;  
}
```

Output

```
Geeks .  
For .  
Geeks .
```

```
Geeks . \nFor . \nGeeks . \n
```

Time Complexity: $O(n)$

Space complexity: $O(n)$

This article is contributed by **MAZHAR IMAM KHAN**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [write.geeksforgeeks.org](https://www.geeksforgeeks.org/write-geeksforgeeks/) or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks. Please write comments if you find anything incorrect, or if you want to share more information about the topic discussed above.

144

Related Articles

1. How to Print String Literal and Qstring With Qdebug in C++?
2. ASCII NULL, ASCII 0 ('0') and Numeric literal 0
3. Integer literal in C/C++ (Prefixes and Suffixes)
4. Multi-Character Literal in C/C++
5. `std::string::length`, `std::string::capacity`, `std::string::size` in C++ STL
6. `std::string::replace` , `std::string::replace_if` in C++
7. `std::string::replace_copy()`, `std::string::replace_copy_if` in C++
8. `std::string::append` vs `std::string::push_back()` vs Operator `+=` in C++
9. `std::string::remove_copy()`, `std::string::remove_copy_if()` in C++
10. Check if a string can be formed from another string using given constraints