

# User-Defined Literals: Introduction

In this lesson, we will introduce user-defined literals.

## WE'LL COVER THE FOLLOWING ^

- User-Defined Literals
  - Syntax
    - Examples
  - The Magic

## User-Defined Literals #

**User-defined literals** are a unique feature in all mainstream programming languages, since they enable us to combine values with units.



## Syntax #

Literals are explicit values in a program, including a `boolean` like `true`, the number `3` or `4.15`; the `char a`, or the C string `"hello"`. The lambda function `[(int a, int b){ return a+b; }]` is also a function literal.

With C++11, it is possible to generate user-defined literals by adding a suffix to a built-in literal for the `int`, `float`, `char`, and C strings.

User-defined literals must obey the following syntax:

<built\_in-Literal> \_ <Suffix>

Usually, we use the suffix for a unit:

## Examples #

```
101000101_b  
63_s  
10345.5_dm  
123.45_km  
100_m  
131094_cm  
33_cent  
"Hello"_i18n
```

What is the key benefit of user-defined literals? The C++ compiler maps the user-defined literals to the corresponding literal operator, and this literal operator must be implemented by the programmer.

## The Magic #

Let's take a look at the user-defined literal `0101001000_b` which represents a binary value. The compiler maps the user-defined literal `0101001000_b` to the literal operator `operator""_b (long long int bin)`.

A few special rules are important to follow:

- There must be a space between the quotation marks and the suffix.
- The binary value ( `0101001000` ) is in the variable `bin`.
- If the compiler doesn't find the corresponding literal operator, the compilation will fail.

With C++14, we get an alternative syntax for *user-defined types*. They differ from the C++11 syntax because user-defined types in require no space.

Therefore, it is possible to use **reserved keywords** like `_C` as a suffix and use a user-defined literal of the form. This can be seen with `11_C`. The compiler will map `11_C` to the literal `operator""_C (unsigned long long int)`. The simple rule states that we can use suffixes starting with an upper-case letter.

User-defined literals are a very helpful feature in modern C++ if we want to write safety-critical software. Why? Due to the automatic mapping of the user-

defined literal to the literal operator, we can implement type-safe arithmetic.

---

The concept will be further explained with an example in the following lesson.