

The std::basic_string_view Type

std::string_view is built upon the std::basic_string_view type. We'll see the implementation below.

Although we talk about `string_view`, it's important to know that this is only a specialisation of a template class called `basic_string_view`:

```
template<
    class CharT,
    class Traits = std::char_traits<CharT>
> class basic_string_view;
```



`Traits` class is used to abstract the operations on the character type. For example on how to compare characters, how to find one character in a sequence.

Such a hierarchy creates a similar relation as there is for `std::string` which is a specialization of `std::basic_string`.

We have the following specialisations:

Type	Definition
std::string_view	std::basic_string_view
std::wstring_view	std::basic_string_view<wchar_t>
std::u16string_view	std::basic_string_view<char16_t>
std::u32string_view	std::basic_string_view<char32_t>

As you can see, the specializations use a different underlying character type.

For the sake of convenience, in the rest of this section, only `string view` will

For the sake of convenience, in the rest of this section, only `string_view` will be considered.

Head over to the next lesson to learn about `string_view` creation.