General Solution

There are always some steps you can take to ensure that your string_view will work with third party APIs. Find out more below.

If your API supports only null-terminated strings and you cannot switch to a function that takes additional count or size parameter, then you need to convert a view into the string.

For example:

```
#include <iostream>
using namespace std;

void ConvertAndShow(const char *str) {
  auto f = atof(str);
  std::cout << f << '\n';
}


int main() {
  std::string number = "123.456";
  std::string_view svNum { number.data(), 3 };
  // ... some code
  std::string tempStr { svNum.data(), svNum.size() };
  ConvertAndShow(tempStr.c_str());
}
```

`ConvertAndShow` only works with null-terminated strings, so the only way we have is to create a temporary string `tempStr` and then pass it to the function.

> If you want to create a string object from `string_view` then remember to use `.data()` and `.size()` so that you refer to the correct slice of the underlying character sequence.

Apart from APIs, `string_view` also works with `constexpr`. More on that in the next lesson.