

Printing With printf()

This is a third party API which may cause problems with `std::string_view`. We'll learn how to use them together correctly.

For example:

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

int main() {
    std::string s = "Hello World";
    std::string_view sv = s;
    std::string_view sv2 = sv.substr(0, 5);
    printf("My String %s", sv2.data()); // oops?
}
```



Instead you should use:

```
printf("%.5s\n", static_cast<int>(sv2.size()), sv2.data());
```

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

int main() {
    std::string s = "Hello World";
    std::string_view sv = s;
    std::string_view sv2 = sv.substr(0, 5);
    printf("%.5s\n", static_cast<int>(sv2.size()), sv2.data());
}
```



.* - describes the precision, see in the [printf specification](#):

The precision is not specified in the format string, but as an additional

integer, value argument preceding the argument that has to be formatted.

Next, we'll examine the compatibility of `string_view` with `atoi`/`atof`.