

Conversion Functions Like atoi()/atof()

This lesson explains why conversion using atof() is not possible with string_view.

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

int main() {
    std::string number = "123.456";
    std::string_view svNum { number.data(), 3 };
    auto f = atof(svNum.data()); // should be 123, but is 123.456!
    std::cout << f << '\n';
}
```



atof takes only the pointer to a null-terminated string, so **string_view** is not compatible.

To fix this, you might have a look at `from_chars` functions (also added in C++17)

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

int main() {
    // use from_chars (C++17)
    std::string number = "123.456";
    std::string_view svNum { number.data(), 3 };
    int res = 0;
    std::from_chars(svNum.data(), svNum.data()+svNum.size(), res);
    std::cout << res << '\n';
}
```



Perhaps you can see a pattern in the fixes we use when working with different APIs. Now, you'll be introduced to a general solution which ensures

compatibility with `string_view`.