std::byte

C++ now allows us to convert data types into bytes using std::byte.

C++17 is a significant update for the language, and it brings a lot of features in the Standard Library. So far this course has covered the most important aspects, but there are many more things to describe!

`std::byte` is a small type that gives you a view of bytes and bits rather than numeric/char values (like unsigned char).

In the Standard it's defined as `enum`:

```
enum class byte : unsigned char {} ; // in <cstddef>
```

You can initialize `byte` from **unsigned char** with the syntax `byte b{unsigned char}`, which is in fact, another handy C++17 feature that allows you to init a scoped enum with the underlying type[^enumunder].

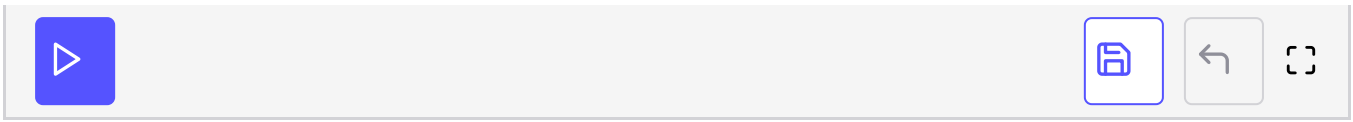
To convert from `byte` into a numeric type use `std::to_integer()`.

> **[^enumunder]:** Read more in P0138 - https://wg21.link/P0138

Let's see a basic example:

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

int main() {
  constexpr std::byte b{1};
  if(std::to_integer<int>(b) == 0x01){
    cout << "b is the byte form of 0x01" << endl;
  }

  // shifts:
  constexpr auto b1 = b << 7;
  if(std::to_integer<int>(b1) == 0x80){
    cout << "b1 is the byte form of 0x80" << endl;
```

```
  }

  // std::byte c{3535353}; // error: narrowing conversion from int
  constexpr std::byte c{255};
  if(std::to_integer<int>(c) == 0xff){
    cout << "c is the byte form of 0xff" << endl;
  }

  // various bit operators, like &, |, &, etc
  constexpr auto c1 = b1 ^ c;
  if(std::to_integer<int>(c1) == 0x7f){
    cout << "c1 is the byte form of 0x7f" << endl;
  }

  constexpr auto c2 = ~c1;
  if(std::to_integer<int>(c2) == 0x80){
    cout << "c2 is the byte form of 0x80" << endl;
  }

  if(c2 == b1){
    cout << "c2 == b1" << endl;
  }

  /*static_assert(std::to_integer<int>(b) == 0x01);
  static_assert(std::to_integer<int>(b1) == 0x80);

  // various bit operators, like &, |, &, etc
  constexpr auto c1 = b1 ^ c;
  static_assert(std::to_integer<int>(c) == 0xff);
  static_assert(std::to_integer<int>(c1) == 0x7f);

  constexpr auto c2 = ~c1;
  static_assert(std::to_integer<int>(c2) == 0x80);
  static_assert(c2 == b1);*/

}
```



The primary motivation behind `std::byte` is type-safety in the context of memory/byte access.

> **Extra Info**: See the reference paper P0298R3.

The next lesson will highlight the additions which have been made to `map` and `set` functionality.