constexpr if

Let's study constexpr if in this lesson.

constexpr if (C++17)

`constexpr if` enables us to compile source code conditionally.

```
if constexpr (cond) statement1;
else statement2;
```

- The expression `cond` has to be a constant expression.
- The unused code has to be valid.
- Thanks to `constexpr if`, functions can have different return types.

The following code snippet shows a function, which returns an `int` or a `double`.

```cpp
template <typename T>
auto getAnswer(T t){
  static_assert(std::is_arithmetic_v<T>); // arithmetic
  if constexpr (std::is_integral_v<T>)    // integral
    return 42;
  else
    return 42.0;        // floating point
}
```

`constexpr if` is a replacement for tag dispatch and SFINAE. SFINAE stands for **S**ubstitution **F**ailure **I**s **N**ot **A**n **E**rror.

To study further about tag dispatching, click here.

In the next lesson, we'll look at an example of `constexpr if`.