# Why do we need attributes?

In this part we learn about attributes!

WE'LL COVER THE FOLLOWING



## Let's start with an example #

Have you ever used `__declspec, __attribute` or `#pragma` directives in your code?

Here is an example:

```
// set an alignment
struct S { short f[3]; } __attribute__ ((aligned (8)));
// this function won't return
void fatal () __attribute__ ((noreturn));
```

Or for DLL import/export in MSVC:

```
#if COMPILING_DLL
#define DLLEXPORT __declspec(dllexport)
#else
#define DLLEXPORT __declspec(dllimport)
#endif
```

Those are existing forms of compiler specific attributes/annotations.

## What is an attribute? #

An attribute is an additional information that can be used by the compiler to produce code. It might be utilized for optimization or some specific code

generation (like DLL stuff, OpenMP, etc.). In addition, annotations allow you to write more expressive syntax and help other developers to reason about code.

Contrary to other languages such as C#, in C++ that the compiler fixes meta information, you cannot add user-defined attributes. In C# you can 'derive' from `System.Attribute`.

## What's best about Modern C++ attributes? #

Since C++11, we get more and more standardised attributes that will work with other compilers. We're moving away from compiler-specific annotation to standard forms. Rather than learning various annotation syntaxes you'll be able to write code that is common and has the same behaviour.

---

In the next section, you'll see how attributes used to work before C++11.