

explicit(boolean-expression)

C++17

```
struct MyInt
{
    int val = 0;

    // explicit if the int type coming in
    // is bigger than what we can hold.
    // eg explicit if you pass a long,
    // implicit if you pass a short
    // TODO: handle unsigned...
    template <typename Init,
        std::enable_if_t<
            (sizeof(Init) > sizeof(int)),
            int> = 0>
    explicit MyInt(Init init)
        : val(static_cast<int>(init))
    {
    }

    template <typename Init,
        std::enable_if_t<
            (sizeof(Init) <= sizeof(int)),
            int> = 0>
    MyInt(Init init)
        : val(static_cast<int>(init))
    {
    }
};
```

C++20

```
struct MyInt
{
    int val = 0;

    // explicit if the int type coming in
    // is bigger than what we can hold.
    // eg explicit if you pass a long,
    // implicit if you pass a short
    // (TODO: check for unsigned...)
    template <typename Init>
    explicit(sizeof(Init) > sizeof(int))
    MyInt(Init init)
        : val(static_cast<int>(init))
    {
    }
};
```

That can be useful on its own. But an interesting corollary is “explicit(false)”. What’s the point of that over the “equivalent” of just not saying explicit at all? It is being explicit about an implicit constructor. For example:

Questionable C++

```
struct Foo
{
    Foo(Bar b);
};
```

Better C++

```
struct Foo
{
    /* yes, implicit*/ Foo(Bar b);
};
```

C++20

```
struct Foo
{
    explicit(false) Foo(Bar b);
};
```

In code review of the first example, I would probably ask if the constructor should be explicit - the general rule is that most constructors should be explicit (it is another default that C++ got wrong). (because implicit

conversions often lead to mistaken conversions, which leads to bugs) So some people leave comments, to make it clear. But comments that just explain syntax (and not *why*) are better done as more clear code.

Using `explicit(false)` shows that the implicitness wasn't an accidental omission, but a real design decision. (You might still want to leave a comment as to *why* that is the right design decision.)

P.S. Please don't `#define implicit explicit(false)`. *Just Don't*. Do Not. Not Do. Don't even Try.

See also wg21.link/p0892