Introduction

The intro to std::any elaborates why and how it is used(explained in later modules)!

WE'LL COVER THE FOLLOWING

- Why use std::any?
- What you'll learn in this Module:

Why use std::any?

With `std::optional` you can represent a regular Type values or mark it as empty. With `std::variant` you can wrap several type alternatives into one entity.

> C++17 gives us one more wrapper type: `std::any` which can hold anything in a `type-safe` way.

What you'll learn in this Module:

- Why `void*` is a very unsafe pattern
- `std::any` and its basic usage
- `std::any` use cases with examples
- `any_cast` and how to use all its "modes"

Let's get started with the basics.