

## Introduction to `std::format`

A new formatting library with C++ 20

# Background & Motivation

- Don't use C-style printf, just don't!!
  - Not type safe
  - Supports only a few numerical types
  - Difficult to localize
  - But has a nice use syntax, hence its popularity
- C++ I/O streams
  - Type safe
  - Extensible through overloading `operator<<`
  - Difficult to localize
  - Verbose syntax

# `std::format`

- Header file: `<format>`
- Simple format syntax; in fact, based on Python's format
  - Separate format string and arguments
- Typesafe, compile-time validation
- Support for positional parameters
- Extensible with user-defined types
- Easy to localize
- Better performance and smaller code size compared to both `printf` and `<iostream>`

# std::format

```
std::string name = ...; // get from somewhere
std::cout << std::format("Hi {}, from std::format!", name);
```

- Use placeholders { }
- Order of declaration is order of use
- But can have positional placeholders

```
std::string name = ...; // get from somewhere
int age = ...; get from somewhere
std::cout << std::format("My name is {0}, my age is {1}", name, age);
```

- Use of numeric formatting

```
std::cout << std::format("{} , the value of Pi is {:.4f}\n",
    name,
    std::numbers::pi_v<float>);
```

# std::format

- Format specifier: `[fill][align][sign][#][width][.precision][type]`
  - **fill**: specify fill character
  - **align**: < (left) > (right) ^ (center)
  - **sign**: -, +, space (show – for neg numbers, space for pos)
    - Not the word ‘space’, an actual single space character
  - **#**: enable alternate formatting rules (hex, binary, octal, etc)
  - **width**: min field width
  - **precision**: number of digits or characters of a string
  - **type**: b (binary), d (decimal), o (octal), x (hex), and more
- There is a lot more you can do with this, including using dynamic values in these fields; e.g., dynamically specify precision
- Formatting spec description...
  - [https://en.cppreference.com/w/cpp/utility/format/formatter#Standard\\_format\\_specification](https://en.cppreference.com/w/cpp/utility/format/formatter#Standard_format_specification)