

ratio 템플릿

- 컴파일 시간 분수 값을 나타내는 템플릿
- `<ratio>` 헤더
- 2개의 static member data 로 구성된다.

| | |
|------------|-------------------------------|
| num | 분자(numerator)를 나타내는 컴파일 시간 상수 |
|------------|-------------------------------|

| | |
|------------|---------------------------------|
| den | 분모(denominator)를 나타내는 컴파일 시간 상수 |
|------------|---------------------------------|

- 분자, 분모는 컴파일 시간 연산을 통해서 약분된 상태로 저장 된다.
- ratio 자체가 실행 시간에 메모리에 보관하는 값은 없고, 오직 컴파일 시간에 사용되는 상수 값이다.

compile – time ratio 연산

| | | |
|------------|----------------|---------------------|
| Arithmetic | ratio_add | ratio_subtract |
| | ratio_multiply | ratio_divide |
| Comparison | ratio_equal | ratio_not_equal |
| | ratio_less | ratio_less_equal |
| | ratio_greater | ratio_greater_equal |

constants

- yocta, zepto, atto, femto, **pico, nano, micro, milli, centi**, deci
- deca, hector, **kilo, mega, giga, tera**, peta, exa, zetta, yotta

duration

- <chrono> 헤더
- namespace **chrono**;
- ratio 로 표현되는 **단위(주기)에 대한 값을 보관하는 클래스**
- **오직 하나의 값을** 보관한다.
- duration을 사용하면 단위에 맞게 자동으로 연산이 수행된다.



캐스팅

| | |
|----------------------------|--|
| <code>duration_cast</code> | Converts a duration to another |
| <code>floor</code> | rounding down, C++17 |
| <code>ceil</code> | rounding up, C++17 |
| <code>round</code> | rounding to nearest, ties to even, C++17 |
| <code>abs</code> | absolute value, C++17 |

시간 관련 타입

```
using nanoseconds = duration<long long, nano>;  
using microseconds = duration<long long, micro>;  
using milliseconds = duration<long long, milli>;  
using seconds = duration<long long>;  
using minutes = duration<int, ratio<60, 1>>;  
using hours = duration<int, ratio<3600, 1>>;
```

초기화 방법

- explicit 생성자 이므로 **direct initialization** 만 사용가능
- 시간 관련 **user define literal** 사용 - 복사 생성

operator ""**h**

operator ""**min**

operator ""**s**

operator ""**ms**

operator ""**us**

operator ""**ns**

time_point

- 기간의 시작과, 경과 개수를 나타내는 타입
- **epoch time** : 1970년 1년 1일 0시를 기점으로 경과된 시간단위