

Day-34

constexpr, constinit, if constexpr

constexpr Keyword

- **constexpr** 또는 immediate functions(즉시 실행 함수)는 함수의 컴파일 타임 평가를 보장함
- 상수가 아닌 값은 항상 컴파일 오류를 생성함

```
constexpr int square(int value){  
    return value * value;  
}
```

```
square(4); // 컴파일 타임 평가
```

```
int v = 4; // "v"는 dynamic  
// square(v); // 컴파일 에러
```

constinit Keyword

- **constinit**은 변수의 컴파일 타임 초기화를 보장함
- 상수가 아닌 값은 항상 컴파일 에러를 생성함

```
constexpr int square(int value){  
    return value * value;  
}
```

```
constinit int v1 = square(4);    // 컴파일 타임 평가  
v1 = 3;                          // ok, v1 변경 가능
```

```
int a = 4; // "v"는 dynamic
// constinit int v2 = square(a); // 컴파일 에러
```

if constexpr

- `if constexpr`은 컴파일 시간 값에 따라 조건부로 코드를 컴파일할 수 있는 C++17 기능
- `if constexpr` 문은 컴파일러가 컴파일 타임에 분기를 평가하도록 강제함(`#if` 전처리기와 유사하게)

```
auto f(){
    if constexpr (sizeof(void*) == 8)
        return "hello"; // const char*
    else
        return 3;        // int, 컴파일 되지 않음
}
```

if constexpr 함정

- `if constexpr`은 명시적인 `if/else` 문에서만 작동함

```
auto f1(){
    if constexpr (my_constexpr_fun() == 1)
        return 1;
    // return 2.0; // 컴파일 에러, constexpr의 일부가 아님
}
```

- `else if` 분기는 `constexpr`이 요구됨

```
auto f2(){
    if constexpr (my_constexpr_fun() == 1)
        return 1;
    else if(my_constexpr_fun() == 2) // -> else if constexpr
        // return 2.0; // 컴파일 에러 // constexpr이 필요함
    else
        return 3L;
}
```