

Basic_Concepts_3 - if statement | for and while Loops | Range-based for Loop

if

if 실행문의 첫 번째 분기 if는 명시된 조건이 true일때 평가하고, 두 번째 분기는 그 외의 경우에 평가된다.

for and while Loops

```
// 몇 번 반복해야 할지 알고있을 때
for ([초기화]; [조건]; [증감]) {
    ...
}

// 몇 번 반복해야 할지 모를 때
while (조건) {
    ...
}

// 몇 번 반복해야 할지 모르지만 최소 한 번은 반복해야할 때
do {
    ...
} while (cond);
```

for 문의 초기화에서 여러개의 변수를 선언할 수 있으며, 초기화, 조건, 증감 부분을 비워서 무한루프를 돌 수 있도록 구현이 가능하다.

Range-based for Loop

C++11부터 전통적인 for루프 구조를 간소화한 범위 기반 for 루프를 제공한다. for 루프 연산과 동등한 값의 범위를 연산하지만 안전하다.

범위 기반 for루프는 유저가 시작, 끝, 종료를 명시하는것을 피한다. (사람이 직접 명시하면서 생기는 휴먼에러로부터 안전하다는 의미인듯 하다)

범위 기반 for루프는 세 가지 경우에서 응용할 수 있다.

1. 고정 범위 배열 `int array[3]`, `"abcd"`
2. 브랜치 초기화 리스트 `{1, 2, 3}`
3. 어느 오브젝트의 `begin()`, `end()` 함수

```
std::vector vec{1, 2, 3, 4};

for(auto x : vec) {
    cout << x << ", ";
// 출력 : "1, 2, 3, 4"

int matrix[2][4];
for (auto& row : matrix) {
    for( auto element : row)
        cout << "@";
    cout << "\n";
}
// 출력 : @@@@
//          @@@@
```

C++17에선 구조체도 바인딩할 수 있도록 확장되었다.

```
struct A {
    int x;
    int y;
};

A array[] = { {1, 2}, {5, 6}, {7, 1} };
for (auto [x1, y1] : array)
    cout << x1 << ", " << y1 << " "; // 출력 : 1,2 5,6 7,1
```