```
if 문
```

```
기본 구조:
if (조건) {
 // 조건이 참일 때 실행
} else if (조건 2) {
 // 조건 2 가 참일 때 실행
} else {
 // 모든 조건이 거짓일 때 실행
}
설명:
- 조건식은 true 또는 false 로 평가되는 논리식이어야 함.
- else if 를 통해 여러 조건을 순차적으로 검사 가능.
- C++17 부터는 if 조건에서 변수 선언 가능.
예시:
int num = 10;
if (num > 0) {
 std::cout << "양수입니다";
} else if (num == 0) {
 std::cout << "0 입니다";
} else {
 std::cout << "음수입니다";
}
기타:
- if 문 안에 또 다른 if 문을 중첩하여 작성할 수 있음.
- 코드 가독성을 위해 중괄호 {}는 생략하지 않는 것이 좋음.
```

```
switch 문
```

```
기본 구조:
switch (변수) {
 case 값 1:
   // 실행 코드
   break;
 case 값 2:
   // 실행 코드
   break;
 default:
   // 모든 case 에 해당하지 않을 때 실행
}
설명:
- 변수는 정수형, 열거형, char, enum class (C++17 이후) 등 가능.
- break 문을 넣지 않으면 다음 case 로 "fall-through"됨.
- default 는 선택 사항이며, 아무 case 에도 해당하지 않을 때 실행됨.
예시:
char grade = 'B';
switch (grade) {
 case 'A':
   std::cout << "우수";
   break;
 case 'B':
 case 'C':
   std::cout << "좋음";
   break;
 default:
   std::cout << "재수강 권장";
}
기타:
- 여러 case 에 같은 코드를 실행시키고 싶다면 break 없이 이어서 작성.
```

## for 문

```
기본 구조:
for (초기화; 조건; 증감) {
 // 반복할 코드
}
설명:
- 초기화: 반복 시작 전에 한 번 실행.
- 조건: 반복할지 여부를 판단. 거짓이면 반복 종료.
- 증감: 반복할 때마다 실행되는 부분.
예시:
for (int i = 0; i < 5; ++i) {
 std::cout << i << " ";
}
범위 기반 for 문 (C++11~):
std::vector<int> v = \{1, 2, 3\};
for (int x : v) {
 std::cout << x << " ";
}
기타:
- 중첩 for 문을 통해 이중 반복 가능.
- continue, break 등을 통해 루프 제어 가능.
```

## while 문

```
기본 구조:
while (조건) {
 // 조건이 참인 동안 반복
}
설명:
- 조건이 처음부터 거짓이면 한 번도 실행되지 않음.
- 반복 전에 조건을 확인하므로 안전한 루프.
예시:
int count = 0;
while (count < 5) {
 std::cout << count << " ";
 ++count;
}
do-while 문과의 차이:
do {
 // 최소 한 번 실행됨
} while (조건);
기타:
- 무한 루프는 while (true)로 작성 가능.
- 적절한 조건 갱신이 없으면 무한 루프 발생 가능.
```